



**UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**A TORRE DE MANAUS E OUTROS JOGOS DE TORRES: PROPRIEDADES EM
GRAFOS E ALGORITMOS**

Lia Alessandra da Silva Martins

Outubro de 2024

Manaus - AM



**UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**A TORRE DE MANAUS E OUTROS JOGOS DE TORRES: PROPRIEDADES EM
GRAFOS E ALGORITMOS**

Lia Alessandra da Silva Martins

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Informática do Instituto de Computação Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Mestre em Informática.

Orientadora: Rosiane de Freitas Rodrigues, Dra.

Co-orientador: Jonas Costa Ferreira da Silva, Dr.

Outubro de 2024

Manaus - AM

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

M386t Martins, Lia Alessandra da Silva
A Torre de Manaus e outros jogos de torres: propriedades em grafos e algoritmos / Lia Alessandra da Silva Martins . 2024
98 f.: il. color; 31 cm.

Orientadora: Rosiane de Freitas Rodrigues
Coorientador: Jonas Costa Ferreira da Silva
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Algoritmos. 2. Caminhos. 3. Complexidade computacional. 4. Jogos Combinatórios. 5. Teoria dos Grafos. I. Rodrigues, Rosiane de Freitas. II. Universidade Federal do Amazonas III. Título



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Informática

FOLHA DE APROVAÇÃO

"A TORRE DE MANAUS E OUTROS JOGOS DE TORRES: PROPRIEDADES EM GRAFOS E ALGORITMOS"

LIA ALESSANDRA DA SILVA MARTINS

DISSERTAÇÃO DE MESTRADO DEFENDIDA E APROVADA PELA BANCA EXAMINADORA CONSTITUÍDA PELOS PROFESSORES:

Prof. Dr. Rosiane de Freitas Rodrigues - PRESIDENTE

Prof. Dr. Edleno Silva de Moura - MEMBRO INTERNO

Prof. Dr. Leandro Miranda Zatesk - MEMBRO EXTERNO

Profa. Dra. Simone Dantas de Souza - UFF - MEMBRO EXTERNO

MANAUS, 14 de outubro de 2024.



Documento assinado eletronicamente por **Rosiane de Freitas Rodrigues, Professor do Magistério Superior**, em 28/10/2024, às 13:03, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Edleno Silva de Moura, Professor do Magistério Superior**, em 29/10/2024, às 15:42, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de](#)



Documento assinado eletronicamente por **Simone Dantas de Souza, Usuário Externo**, em 29/10/2024, às 16:05, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Leandro Miranda Zatesko, Usuário Externo**, em 30/10/2024, às 12:32, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Maria do Perpétuo Socorro Vasconcelos Palheta, Secretária em exercício**, em 31/10/2024, às 17:16, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2288075** e o código CRC **E149C107**.

Avenida General Rodrigo Octávio, 6200 - Bairro Coroado I Campus Universitário
Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) 3305-1181 / Ramal 1193
CEP 69080-900, Manaus/AM, coordenadorppgi@icomp.ufam.edu.br

Referência: Processo nº 23105.043911/2024-34

SEI nº 2288075

Dedicatória

À minha mãe Paula do Carmo da Silva Martins (in memoriam) por ter sonhado em ser Mestre e ter partido sem tê-lo concretizado. Por ter me pedido tanto para continuar e fazer o Mestrado. Por ter me mostrado o quão importante é estudar. Obrigada por ter me deixado seguir, mesmo quando sua maior vontade era pedir para eu ficar. Obrigada por ter dedicado sua vida a mim. Obrigada por ter sido a mãe mais incrível que eu poderia ter tido. Obrigada por estar sempre ao meu lado. Sempre foi e sempre vai ser tudo por você e para você!

Agradecimentos

À Deus, por ter me dado forças, por ter permitido eu chegar até aqui e ter colocado pessoas maravilhosas em minha vida.

Aos meus pais, Garcegilson Pontes Martins (in memoriam) e Paula do Carmo da Silva Martins (in memoriam), por não terem medido esforços para que eu pudesse estudar.

Aos meus irmãos, Leonardo Martins e Lua Aymê, os quais eu tenho a honra de amar e cuidar como filhos. Obrigada por muitas vezes serem meus braços. Obrigada por todo amor, apoio, carinho e por entenderem os momentos que precisei estar ausente. Amo vocês mais que tudo!

Ao meu amor, Lanier Menezes dos Santos, por estar ao meu lado sempre, me apoiando e incentivando. Obrigada por me amar, cuidar e compreender.

À minha orientadora, Rosiane de Freitas, principalmente por nunca ter desistido de mim. Por nunca me olhar como uma doente. Por acreditar em mim, quando todos os médicos disseram que eu não podia. Por todos os ensinamentos a mim repassados.

Ao meu co-orientador, Jonas Costa, o anjo que papai do céu mandou. Obrigada principalmente por toda paciência ao me ensinar. Obrigada por todo apoio e incentivo.

Aos meus tios, Selma Brandão (in memoriam), Geberson Brandão, Samara Brandão, Carliene Cavalier e Telma Lúcia, por me amarem, por me cuidarem, por me incentivarem, por todos os carinhos, por todas as massagens e por todo apoio que sempre me deram nesta caminhada.

Aos meus sogros, Raimunda Santos e Lenoir Santos, pelo apoio e incentivo no decorrer deste processo, foram inúmeras vezes os meus braços e pernas, me acompanharam inclusive nas idas na UFAM. Nunca vou esquecer tudo que fizeram por mim.

Aos meus companheiros de caminhada, Thailsson Clementino, Meng Hsu, João Bessa, Larissa Pessoa, Victoria Guimarães, Raquel Folz, Reine Santos e Jesse Deveza, por me ouvirem quando batia o desespero, por me ajudarem quando eu não fazia ideia de como seguir, por me motivarem e por todo conhecimento compartilhado.

“E você aprende que realmente pode suportar, que realmente é forte, que pode ir muito mais longe depois de pensar que não se pode mais. E que realmente a vida tem valor e que você tem valor diante da vida”. (William Shakespeare)

Resumo da Dissertação apresentada à IComp/UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Informática (M.Sc.)

A TORRE DE MANAUS E OUTROS JOGOS DE TORRES: PROPRIEDADES EM GRAFOS E ALGORITMOS

Lia Alessandra da Silva Martins

Outubro/2024

Orientadora: Rosiane de Freitas Rodrigues

Co-orientador: Jonas Costa Ferreira da Silva

Nesta dissertação é proposto um novo jogo baseado em torres, o Jogo da Torre de Manaus (ToM), sendo uma variação do jogo clássico da Torre de Hanoi (ToH), com restrições nas capacidades dos pinos. Assim, dada uma estrutura com 3 pinos e n discos, o primeiro pino suporta todos os n discos, enquanto o segundo e o terceiro pinos suportam no máximo $n - 1$ e $n - 2$ discos, respectivamente. O objetivo do jogo consiste em, partindo de uma configuração inicial aleatória, mover todos os discos para o primeiro pino respeitando-se tanto as restrições de capacidade quanto as regras do jogo da ToH, de sempre mover um disco por vez e nunca colocar um disco maior em cima de um menor. Dependendo da configuração inicial, o objetivo pode não ser alcançável, o que resulta em um modelo de grafos não conexo. Assim, é dada a caracterização do grafo da ToM, que modela todas as configurações possíveis do jogo como vértices, e as transições válidas entre elas como arestas. São exploradas propriedades desse grafo, como o número de vértices e de componentes conexas. É, também, proposto um algoritmo para solucionar o jogo quando possível, apresentando-se para isto um algoritmo para decidir se o objetivo é alcançável a partir de uma dada configuração ou não. Adicionalmente, é apresentada uma compilação de modelagem em grafos e aspectos algorítmicos das principais variações da ToH encontradas na literatura. Dentre elas, as Torres de Londres e de Oxford, que eram conhecidas apenas como testes neurocognitivos, são formalizadas neste trabalho como jogos combinatórios. Por fim, é proposta uma formalização da construção recursiva do grafo da ToH, usando-se apenas elementos de Teoria dos Grafos, o que ainda não se conhecia na literatura.

Palavras-chave: algoritmos, caminhos, complexidade computacional, jogos combinatórios, teoria dos grafos, Torre de Hanoi.

Abstract of Dissertation presented to IComp/UFAM as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

THE TOWER OF MANAUS AND OTHER TOWER PUZZLES: PROPERTIES IN
GRAPHS AND ALGORITHMS

Lia Alessandra da Silva Martins

October/2024

Advisor: Rosiane de Freitas Rodrigues

Co-advisor: Jonas Costa Ferreira da Silva

In this research, the puzzle game Tower of Manaus (ToM) is proposed, which is a variation of the classic Tower of Hanoi (ToH) with capacity constraints on the pegs. Thus, given a structure with 3 pegs and n disks, the first peg supports all n disks, while the second and third pegs support at most $n - 1$ and $n - 2$ disks, respectively. The goal is, starting from an arbitrary initial configuration, moving all disks to the first peg, respecting both the capacity constraints and the ToH game rules, always moving one disk at a time and never placing a larger disk on top of a smaller one. Depending on the initial configuration, the goal may not be achievable, which results in a nonconnected graph. Thus, the characterization of the ToM graph is given, which models all possible configurations of the game as vertices, and the valid transitions between them as edges. Properties of this graph, such as the number of vertices and connected components, are explored. An algorithm to solve the game when possible is also proposed, presenting an algorithm to decide whether the goal is achievable from a given configuration or not. Additionally, a compilation of graph modeling and algorithmic aspects of the main variations of ToH found in the literature is presented. Among them, the Towers of London and Oxford, which were known only as neurocognitive tests, are formalized as combinatorial games. Finally, a formalization of the recursive construction of the ToH graph is proposed, using only elements of Graph Theory, which was not yet known in the literature.

Keywords: algorithms, combinatorial games, computational complexity, graph theory, paths, Tower of Hanoi.

Sumário

Lista de Figuras	1
Lista de Tabelas	6
Lista de Algoritmos	7
1 Introdução	8
2 Fundamentação Teórica	12
2.1 Jogos Combinatórios de um Jogador	12
2.2 Conceitos Importantes em Teoria dos Grafos	14
2.3 Algoritmos e Aspectos de Complexidade Computacional	17
3 O Jogo da Torre de Hanoi	20
3.1 O Jogo, Estrutura e Regras	20
3.2 Aspectos Algorítmicos	23
3.3 Formas de Representação de um Estado do Jogo	25
3.3.1 Rótulo de um Estado	25
3.4 Grafo da Torre de Hanoi Baseado no Triângulo de Sierpinski	26
3.5 Aplicações da Torre de Hanoi	30
4 Outros Jogos e Testes Baseados em Torres	32

4.1	Classificação por Características Estruturais e Jogabilidade	32
4.2	Torre de Bucareste	34
4.2.1	Aspectos Algorítmicos	35
4.2.2	Modelagem em Grafos	36
4.3	Torre de Londres	38
4.3.1	Aspectos Algorítmicos	40
4.3.2	Modelagem em Grafos	43
4.4	Torre de Oxford	44
4.4.1	Aspectos Algorítmicos da Torre de Oxford Versão 1	47
4.4.2	Aspectos Algorítmicos da Torre de Oxford Versão 2	47
4.4.3	Modelagem em Grafos	49
5	Proposta do Jogo Torre de Manaus	53
5.1	Unindo Características dos Jogos da Torre de Hanoi e da Torre de Londres	53
5.2	Aspectos Algorítmicos	58
5.3	Modelagem em Grafos	60
5.4	Aplicações da Torre de Manaus	71
5.5	Trabalhos Relacionados	71
6	Outros Resultados	72
6.1	Modelagem e Generalização do Grafo da Torre de Hanoi por Estruturas em Grafos	72
6.1.1	Operação de Enxerto	73
6.1.2	Construção do Grafo da Torre de Hanoi usando Enxertos em Ciclos C_6	75
6.1.3	Similaridades do Grafo da Torre de Hanoi e o Cobra Triangular Alternada	77

6.2	Outras Contribuições Envolvendo os Jogos de Torres Explorados	79
6.3	Análise Comparativa sobre os Grafos das Variações dos Jogos de Torres	
	Explorados	80
7	Considerações Finais	83
	Apêndices	90
	Apêndice A Representação Binária	90
	Apêndice B Código Gray	92
	Apêndice C Triângulo de Sierpinski	94
	Apêndice D Problemas de Pilha na Gestão de Logística	96
	Apêndice E Problemas de <i>Backup</i> na Área de TI	98

Lista de Figuras

1.1	Visão geral da metodologia científica aplicada.	10
2.1	Exemplo de grafo G em (a) e subgrafos de G em (b) e (c).	14
2.2	Exemplos de grafos: em (a) um grafo completo com 5 vértices - K_5 , em (b) um grafo caminho com 4 vértices - P_4 e em (c) um grafo ciclo com 4 vértices - C_4 .	15
2.3	Grafos isomorfos G e H .	16
2.4	Tipos de emparelhamento: em (a) um exemplo de emparelhamento máximo e em (b) um exemplo de emparelhamento perfeito, onde todos os vértices são saturados por M .	16
2.5	Exemplo da classe de grafo Cobra-triangular - T_5 .	17
2.6	Exemplo da classe cobra-triangular alternada - A_6 .	17
3.1	Lenda que inspirou a estrutura e regras do jogo da Torre de Hanoi.	20
3.2	Torre de Hanoi: (a) representação tangível do jogo com 3 pinos, com o menor disco, na cor verde, no topo da pilha e o maior disco, na cor azul, na base; (b) demonstração da solução do jogo com 3 discos, onde os discos são movidos de p_0 para p_2 , até que o objetivo do jogo seja alcançado.	21
3.3	Torre de Hanoi com 1 disco: (a) estados possíveis do jogo; (b) a representação gráfica desses pontos (estados) no grafo, formando assim, o grafo H_3^1 .	28
3.4	Torre de Hanoi com 2 discos: em (a) todos os estados possíveis do jogo e em (b) a representação gráfica do grafo H_3^2 .	28

3.5 Torre de Hanoi com 3 discos: em (a) todos os estados possíveis do jogo e em (b) a representação gráfica do grafo H_3^3 .	29
3.6 H_3^3 : Caminho do exemplo da solução do jogo, com o menor número de movimentos, no grafo.	29
4.1 Torre de Bucareste: (a) representação tangível do jogo com 3 pinos, com o menor disco, na cor verde, no topo da pilha e o maior disco, na cor azul, na base; (b) demonstração da solução do jogo com 3 discos, onde os discos são movidos de p_0 para p_2 obedecendo as regras, até que o objetivo do jogo seja alcançado.	35
4.2 Grafos da Torre de Bucareste.	37
4.3 Grafo B_3^3 com exemplo de uma jogada com 3 discos, em amarelo o caminho percorrido do estado inicial até o estado final do jogo.	38
4.4 Apresentação da ToL: (a) representação tangível do jogo com 3 pinos de tamanhos diferentes, com 3 esferas de mesmo tamanho, demonstrando a posição inicial do jogo; (b) um exemplo de cartão-problema do jogo.	39
4.5 ToL: demonstração da solução do jogo partindo da posição inicial e seguindo os passos até chegar no estado final determinado pelo cartão-problema.	39
4.6 ToL: possíveis estados iniciais do jogo e o M - número mínimo de movimentos para atingir o estado final do jogo.	40
4.7 ToL: (a) Estado final do jogo definido pela sequência de cores verde, vermelho e azul e (b) demonstração da solução do jogo, mostrando todos os passos, até que o objetivo do jogo seja alcançado.	40
4.8 Grafos da Torre de Londres.	43
4.9 Torre de Londres: exemplo do caminho de uma jogada no grafo L_3^3 .	45
4.10 Apresentação da Torre de Oxford.	46
4.11 Apresentação da Torre de Oxford Versão 1.	46
4.12 Demonstração do jogo da ToO Versão 2.	47

4.13 Grafos da Torre de Oxford.	50
4.14 Grafo O_3^3	51
4.15 ToO: exemplo do caminho percorrido para solucionar o jogo no grafo O_3^3	52
5.1 Torre de Manaus: (a) representação tangível do jogo com 4 discos, com o menor disco, na cor verde, no topo da pilha e o maior disco, na cor amarela, na base; (b) demonstração da solução do jogo com 3 discos, partindo de um estado regular inicial e empilhando todos os discos em ordem decrescente de tamanho em p_0	53
5.2 ToM com 3 discos apresentando todas as possibilidades de estados regulares iniciais do jogo. Sendo que, a coluna M mostra o número mínimo de movimentos necessários para o jogador que iniciar o jogo nesse estado possa solucionar o jogo ou o símbolo vermelho que indica que partindo daquele estado, não é possível solucionar o jogo sem violar as regras.	54
5.3 ToM com 4 discos apresentando todas as possibilidades de estados regulares iniciais do jogo. Sendo que, a coluna M mostra o número mínimo de movimentos necessários para o jogador que iniciar o jogo nesse estado possa solucionar o jogo ou o símbolo vermelho que indica que partindo daquele estado, não é possível solucionar o jogo sem violar as regras.	55
5.4 ToM: em (a) M_3^2 e (b) M_3^3 , de verde os vértices que representam estados bons e de vermelho os vértices que representam estados ruins.	61
5.5 M_3^4 - Grafo da ToM com 4 discos, de verde os vértices que representam os estados bons e de vermelho os vértices que representam estados ruins.	62
5.6 ToM: exemplo de uma sequência de movimentos para ir de um estado ruim em que d_n está em p_2 para o estado $d_1, d_2, \dots, d_{n-1} // d_n$	63
5.7 ToM: outra sequência de movimentos para ir de um estado ruim em que d_n está em p_2 para o estado $d_1, d_2, \dots, d_{n-1} // d_n$	63
5.8 Estrutura do grafo da ToM, onde: TOP - representa o topo; ESQ - o lado esquerdo; e, DIR - o lado direito do grafo.	65
5.9 Exemplo da estrutura de particionamento do Lema 4 no grafo M_3^4	66

5.10	M_3^3 : em amarelo o caminho ótimo a ser percorrido para solucionar o jogo, partindo do estado inicial 1/2/3.	67
5.11	Torre de Manaus com $k + 1$ discos.	70
6.1	Grafo H_3^3 destacando as estruturas (a) C_3 , (b) C_9 e (c) C_6 .	73
6.2	Grafos (a) G e (b) Q para exemplificar a operação de enxerto.	73
6.3	Grafo R , resultante da operação de enxerto, $R = enx(Q, v, w, G, ab)$, do grafo Q , Figura 6.2b, no grafo G , Figura 6.2a.	74
6.4	Operação de enxerto $R' = enx(Q', v, w, G', ab)$, explicitando que a aresta ab somente existirá em $E(R)$ se, somente se, $vw \in E(Q')$.	74
6.5	Ilustração do processo recursivo de construção do Grafo H_3^n (c), onde (a) C_6 destacando as arestas pertencentes ao emparelhamento perfeito, (b) o grafo H_3^{n-1} que será enxertado.	75
6.6	Exemplo do processo recursivo de construção do Grafo H_3^3 (c), onde (a) C_6 em destaque as arestas pertencentes ao emparelhamento perfeito, (b) o grafo H_3^2 que foi enxertado nas arestas do C_6 .	76
6.7	Grafo H_3^2 baseado no Cobra Triangular Alternada $A(T_6)$, em destaque a aresta u_1u_6 que foi adicionada para completar o grafo.	77
6.8	H_3^3 baseado no Cobra Triangular Alternada $A(T_6)$.	78
6.9	H_3^4 baseado no Cobra Triangular Alternada $A(T_6)$.	78
6.10	Grafo H_3^{k+1} composto por $3(H_3^k)$ disjuntos em vértices.	79
6.11	Grafos das torres exploradas: O_3^3 , L_3^3 , H_3^3 e B_3^3 .	81
6.12	Comparação entre os grafos da ToH e ToL: em verde os vértices pertencentes a ambos os grafos, em vermelho os que pertencem apenas ao L_3^3 e em cinza os que pertencem somente ao H_3^3 .	82
6.13	Comparação entre os grafos H_3^3 , L_3^3 e M_3^3 .	82
A.1	Torre de Hanoi com solução binária.	91
B.1	Resolução da Torre de Hanoi com Código Gray.	93

C.1	Primeiros níveis de construção do Triângulo de Sierpinski.	94
C.2	Triângulo de Sierpinski. Fonte: [Eppstein, Frishberg e Maxwell 2022].	95
D.1	Exemplos de pilhas na Gestão de Logística.	96
D.2	Organização de contêineres em um navio.	97
E.1	Exemplo do <i>backup</i> em fitas.	98

Lista de Tabelas

3.1 Torre de Hanoi: cálculo matemático para encontrar o número mínimo de movimentos com 1, 2, 3, 4, 5, ..., n discos.	22
4.1 Principais resultados na literatura.	33
4.2 Torre de Bucarest: cálculo matemático para encontrar o número mínimo de movimentos com 1, 2, 3, 4, 5, ..., n discos.	35
6.1 Análise dos grafos das Torres exploradas.	80
A.1 Contador Binário, onde o número da jogada em binário indica o movimento que deve ser realizado.	91
B.1 Construção do Código Gray.	92

Lista de Algoritmos

1	Torre de Hanoi - Algoritmo Recursivo.	23
2	Torre de Hanoi - Algoritmo Iterativo.	24
3	ToH - 1º Movimento Ótimo de um Estado Abandonado.	25
4	ToH - Detecção de Desvio do Caminho Ótimo.	26
5	ToB - Algoritmo Iterativo.	36
6	Torre de Bucareste - Algoritmo Recursivo.	36
7	ToL - Algoritmo Recursivo	42
8	Torre de Oxford Versão1 - Algoritmo Iterativo.	47
9	ToO - Algoritmo Recursivo	48
10	Torre de Manaus - Identificação do Estado.	58
11	Torre de Manaus - Resolução de Um Estado Bom.	59
12	Torre de Manaus - Algoritmo Recursivo.	60
13	Enxerto de múltiplas cópias de Q	75
14	Construção do grafo H_3^n .	76

Capítulo 1

Introdução

Um conjunto de estados ou configurações, um conjunto de regras ou movimentos possíveis, estado inicial e estado final, são características que definem um jogo como um jogo combinatório [Berlekamp, Conway e Guy 2001]. Geralmente, os jogos combinatórios possuem mais de um jogador, mas podem ser de jogador único [Hearn e Demaine 2009]. As informações do jogo são completas, ou seja, os jogadores têm acesso a todas as informações relevantes sobre o estado do jogo e as ações dos outros jogadores. **Por fim**, os jogadores podem ser perfeitos ou imperfeitos, no sentido de sempre realizarem o melhor movimento possível [Cormen 2017].

Neste trabalho são abordados jogos combinatórios na forma de uma ou mais torres (ou pilhas), como o jogo clássico da Torre de Hanoi (ToH). Os jogos envolvendo torres são quebra-cabeças de raciocínio lógico e normalmente são utilizados para medir habilidades de planejamento e desempenho na execução de tarefas [Milla et al. 2019]. Nesse sentido, esses jogos exigem que o indivíduo resolva um problema específico usando o mínimo de movimentos.

O jogo da ToH, o mais conhecido dentre os jogos baseados em torres, possui uma base contendo três pinos, em um dos pinos, denominado origem, são dispostos discos uns sobre os outros, do maior para o menor. O objetivo do jogo é mover todos os discos do pino origem onde se encontram, para um pino destino, obedecendo a duas regras simples: só pode mover um disco de cada vez e um disco maior não pode ser colocado em cima de outro menor.

Vale ressaltar que estes jogos baseados em torres possuem um potencial enorme

de aplicação em psicologia, como é o caso da Torre de Londres (ToL), uma variação da ToH, que foi projetado para avaliar a capacidade do indivíduo de planejar e solucionar problemas [Such et al. 2011]. A ToL possui 3 esferas de mesmo tamanho, uma base retangular com 3 torres (ou pinos) de tamanhos diferentes: a primeira é a mais longa, suporta 3 esferas, a segunda apenas 2 esferas e a terceira e mais curta das torres, só sustenta 1 esfera. O estado inicial da ToL são 2 esferas na primeira torre e 1 esfera na segunda torre. A ToL vem com uma cartela de cartões-problemas que representam o estado final do jogo. O objetivo da ToL é atingir este estado final (que varia a cada desafio do jogo) com o menor número de movimentos.

Nesta pesquisa foram investigados jogos combinatórios baseados em torres, com o propósito de explorar estratégias algorítmicas, assim como uma nova variação desses jogos, para analisar as estruturas em grafos, conforme essas modificações forem sendo aplicadas. Desse modo, como **questão de pesquisa** desta Dissertação de Mestrado têm-se:

- (1) É possível caracterizar teoricamente uma variação do jogo da Torre de Hanoi clássico que apresente uma jogabilidade diferente, interessante e que resulte em desafios teóricos relacionados à modelagem em grafos e estratégias algorítmicas de resolução? É possível definir propriedades em grafos para esta nova possível versão do jogo clássico?

O **objetivo geral** desta dissertação é caracterizar teoricamente uma variação do jogo da Torre de Hanoi, aplicando ferramental de teoria dos grafos e algoritmos que apresente propriedades matemático-computacionais diferentes do jogo original. Os objetivos específicos são: investigar variações conhecidas do jogo da Torre de Hanoi que diferem pela estrutura do jogo e/ou por inserir/relaxar regras; analisar e caracterizar teoricamente o teste da Torre de Londres e outros possíveis e, finalmente, determinar o modelo em grafos da variação do jogo da Torre de Hanoi proposto.

A visão geral da **metodologia** utilizada para encontrar respostas às questões de pesquisa relacionadas é apresentada na Figura 1.1. Foi desenvolvida uma pesquisa de natureza essencialmente **básica**, com geração de conhecimento teórico sobre o jogo combinatório baseado em torres proposto, tendo como base toda a fundamentação teórica existente para o jogo clássico da Torre de Hanoi.

Em relação ao objeto de pesquisa, inicialmente, foi produzida uma **pesquisa ex-**



Figura 1.1. Visão geral da metodologia científica aplicada.

ploratória, com o intuito de proporcionar uma maior familiaridade com a ToH e as variações já existentes. Isso abrangeu o levantamento bibliográfico e a análise dos jogos combinatórios baseados em torres, a qual foi realizada por meio de revisão da literatura em bases de dados tais como: IEEE, *Springer*, *Scopus* e *Elsevier*. Em seguida, foi praticada uma **pesquisa descritiva** com o propósito de descrever as variações da Torre de Hanoi, caracterizando-as como um jogo, analisando-as computacionalmente e realizando a modelagem em grafos. Por fim, gerou-se uma nova variação do jogo da ToH e realizou-se uma **pesquisa explicativa** onde foram definidos os fenômenos que podem ocorrer nesse novo jogo.

Além disso, foi utilizada a estratégia de construção do conhecimento, por meio do **método dedutivo**. Para tanto, baseou-se nas características e no grafo da ToH e das variações do jogo, tais como: generalizações matemáticas, algoritmos recursivo e iterativo, para elaboração da nova versão que está sendo proposta. Ademais, foi utilizado o **método indutivo** para se apoiar nos conhecimentos obtidos, ao analisar a ToH e as variações estudadas, para elaboração das especificidades do novo jogo.

Como **principal contribuição desta pesquisa** está a proposta do jogo da Torre de Manaus (ToM), um jogo que une as características da Torre de Hanoi (ToH) e a Torre de Londres (ToL). Esse jogo foi definido com restrições, objetivo e um algoritmo recursivo para resolvê-lo. Foi produzido, adicionalmente, o grafo que representa todos os estados do jogo, e a estrutura desse grafo foi caracterizada. Realizou-se, também, a generalização da construção do grafo da ToH usando somente estruturas em grafos, com um algoritmo desenvolvido para sua formação recursiva. Por fim, foram adaptados alguns testes neurocognitivos existentes, tais como: a ToL e a Torre de Oxford (ToO) e definidos como jogos combinatórios.

O restante desta dissertação está estruturada da seguinte maneira: No Capítulo 2, é apresentada a fundamentação teórica necessária ao entendimento deste trabalho. No Capítulo 3, o jogo da ToH é explorado, abordando-se aspectos matemáticos e algorítmicos, a complexidade computacional, modelagem em grafos e, finalizando, com aplicações práticas do jogo. No Capítulo 4, é feita uma compilação da literatura sobre jogos baseados em torres, destacando as principais variações da ToH, além de explorar os algoritmos e a modelagem em grafos. No Capítulo 5, é apresentado o novo jogo denominado Torre de Manaus. No Capítulo 6, são apresentados resultados adicionais desta pesquisa, incluindo a generalização do grafo da ToH usando estruturas em grafos, as contribuições para algumas variações da ToH, e uma análise comparativa das variações discutidas neste trabalho. No Capítulo 7, são tecidas as considerações finais e relacionados possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo se discorre acerca das principais definições básicas necessárias para o bom entendimento deste trabalho. Inicia-se com a teoria dos jogos combinatórios, abordando seus objetivos principais e as condições necessárias para um jogo ser considerado um jogo combinatório. Além disso, são apresentados os principais conceitos em teoria dos grafos e, por fim, é comentado a respeito de algoritmos e complexidade computacional.

2.1. Jogos Combinatórios de um Jogador

A teoria dos jogos pode ser definida como a teoria dos modelos matemáticos que estuda a escolha de decisões ótimas sob condições de conflito. Existem alguns exemplos clássicos de problemas modelados na área de teoria dos jogos, tais como: o dilema dos prisioneiros, jogos com escolhas simultâneas, roteamento por provedores de serviço, batalha dos sexos e cara ou coroa [Miyazawa 2014].

Esta pesquisa foca em jogos combinatórios de um jogador. No entanto, é importante destacar que, na teoria dos jogos [Ferguson 2000], existem termos fundamentais que estruturam e analisam situações estratégicas, entre os quais se identificam:

- **jogo**: um modelo formal, o que significa que a teoria dos jogos envolve descrições e análises técnicas;
- **interações**: as ações individuais de cada agente afetam os outros agentes;
- **agentes**: qualquer indivíduo ou grupo de indivíduos tem capacidade de tomar decisões que afetam outros agentes. O agente no contexto da teoria dos jogos é denominado jogador;

- **racionalidade:** assumindo que todos os agentes são racionais, significa que eles utilizam o método mais adequado buscando a satisfação dos seus desejos;
- **comportamento estratégico:** isso significa que cada jogador leva em consideração o fato de que todos os jogadores interagem entre si. Dessa maneira, a decisão tomada irá gerar consequências aos outros jogadores.

A teoria dos jogos tradicionalmente estuda interações estratégicas entre dois ou mais jogadores, em que cada um procura maximizar seu benefício, considerando as possíveis ações dos outros. Esse campo abrange princípios que não são explorados neste trabalho. No entanto, há uma categoria específica de jogos chamada jogos combinatórios de um jogador. Esses jogos são frequentemente conhecidos como jogos de paciência ou quebra-cabeças. Apesar de não haver competição direta com outro jogador, esses jogos ainda exigem decisões estratégicas baseadas em regras, com o objetivo de alcançar um estado final específico. Assim, podem ser observadas algumas características [Berlekamp, Conway e Guy 2001]:

- **informação perfeita:** o jogador tem acesso a todas as informações necessárias, sabe todas as regras e condições do jogo;
- **movimentos discretos e finitos:** há um número limitado de movimentos possíveis e de estados do jogo;
- **objetivo específico:** o jogador busca alcançar uma condição de vitória ou resolver o desafio, como organizar peças de um certo modo ou limpar o tabuleiro.

Jogos interessantes como o *Sudoku*, o *Solitaire* e os jogos baseados em torres abordados nesta pesquisa, são exemplos de jogos combinatórios de um jogador. Nesses jogos, embora não haja outro jogador, sua natureza combinatória e estratégica faz com que eles se encaixem na categoria de jogos combinatórios, com o jogador competindo essencialmente contra as regras e a estrutura do jogo [Demaine 2001].

Existem, ainda, jogos combinatórios que incluem também os autômatos para nenhum jogador, como o Jogo da Vida de *Conway*. Embora na definição mais estrita, possa-se dizer que jogos exigem mais de um participante, por isso, no caso, as designações de quebra-cabeça e autômato para os casos especiais de um ou nenhum jogador [Hearn e Demaine 2009].

2.2. Conceitos Importantes em Teoria dos Grafos

Esta subseção apresenta os conceitos de teoria dos grafos que orientam a pesquisa e embasam as análises subsequentes, baseando-se nos autores: [Szwarcfiter 2018], [Diestel 2017], [Benjamin, Chartrand e Zhang 2017]. Conceitos mais específicos são referenciados *in loco*.

Um **grafo** $G = (V, E)$ é um conjunto finito não-vazio, onde $V(G)$ é um conjunto de vértices e $E(G)$ é um conjunto de arestas. Dessa maneira, cada elemento e no conjunto E é um par ij que indica que o vértice i é adjacente ao vértice j (ou seja, i e j são adjacentes e a aresta e incide em i e j). Assim, o grafo é dito não-direcionado quando os pares que representam as arestas são não-ordenados, isto é, $ij = ji$. Um exemplo de grafo é mostrado na Figura 2.1a.

O número de vértices de um grafo G é dito **ordem** de G e o número de arestas é o **tamanho** de G . O **grau** de um vértice v do grafo equivale à quantidade de arestas que incidem em v . O grau máximo do grafo, denotado por $\Delta(G)$, é o valor do maior grau dentre todos os vértices de G . De maneira similar, o grau mínimo, denotado por $\delta(G)$, é definido como o valor do menor grau de G .

Um **subgrafo** um grafo H é um subgrafo de um grafo G quando $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Dois exemplos de subgrafos da Figura 2.1a são mostrados nas Figuras 2.1b e 2.1c.

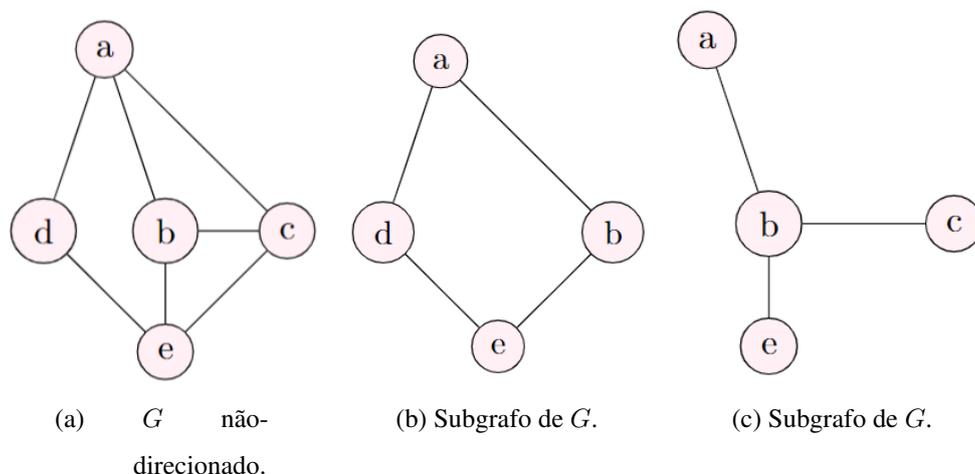


Figura 2.1. Exemplo de grafo G em (a) e subgrafos de G em (b) e (c).

Um **grafo completo**, denotado por K_n , é aquele em que cada vértice é vizinho

de todos os demais, ou seja, cada par de vértices é adjacente entre si, um exemplo é o K_5 apresentado na Figura 2.2a. O grau de todos os vértices de um grafo completo é exatamente igual a $n - 1$. A quantidade de arestas é dada então por $m = \frac{n(n-1)}{2}$.

Um **caminho**, Figura 2.2b, é uma sequência de vértices $P = \{v_1, v_2, \dots, v_n\}$ de tal forma que 2 vértices são adjacentes se e somente se forem consecutivos na sequência. Desse modo, o comprimento ou tamanho de P , denotado por $|P|$, é o número de arestas de P , ou seja, $|P| = n - 1$. Um **ciclo** C_n (para $n \geq 3$) é um grafo simples cujos vértices podem ser dispostos em uma sequência cíclica de tal forma que 2 vértices são adjacentes se são consecutivos na sequência e não adjacentes em caso contrário. O primeiro e último vértices coincidem. Um exemplo de um ciclo C_4 é mostrado na Figura 2.2c. O tamanho de um ciclo é o número de suas arestas. Um ciclo de tamanho k é chamado de k -ciclo.

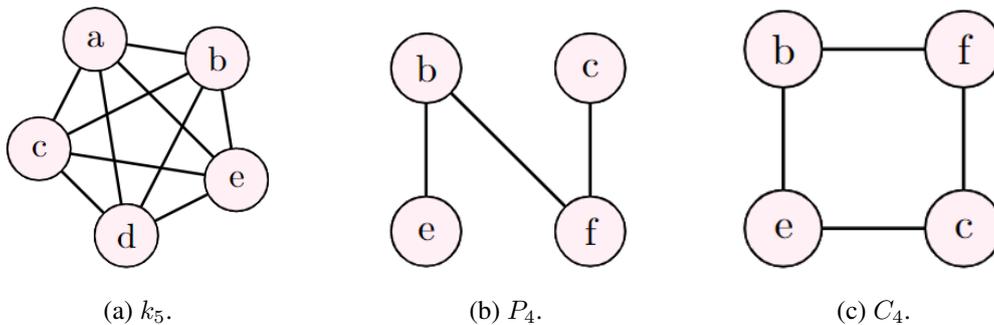


Figura 2.2. Exemplos de grafos: em (a) um grafo completo com 5 vértices - k_5 , em (b) um grafo caminho com 4 vértices - P_4 e em (c) um grafo ciclo com 4 vértices - C_4 .

Um grafo é **conexo** se quaisquer que sejam os vértices distintos u e v de G , existe sempre um caminho que os une. Um grafo que não é conexo é chamado **desconexo**. Um grafo que pode ser representado no plano de forma que suas arestas não se cruzam é chamado de grafo **planar**. Um exemplo de grafo planar é mostrado na Figura 2.1a.

Dois grafos são considerados **isomorfos** se existe uma correspondência um-para-um entre seus vértices e arestas que preserva as conexões entre os vértices, ou seja, possuem a mesma estrutura, mesmo número de vértices e arestas e o mesmo grau de vértices. Formalmente, dois grafos G e H são grafos isomorfos se os vértices de G puderem ser renomeados para produzir H . Por exemplo, os grafos G , Figura 2.3a e H , Figura 2.3b, são isomorfos, pois se renomeados os vértices a, b, c, d e e em G com (respectivamente) 1, 2, 5, 3 e 4 obtém-se H .

Um **emparelhamento** M é um subconjunto de arestas duas a duas não adja-

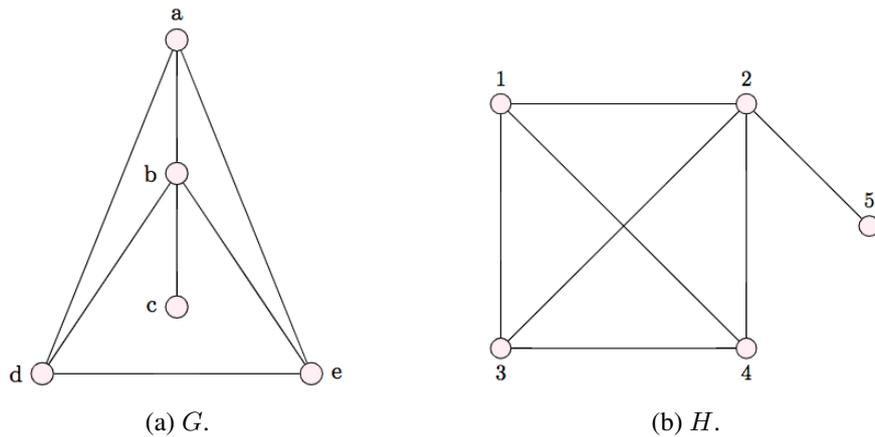


Figura 2.3. Grafos isomorfos G e H .

centes. As arestas pertencentes a M são ditas M - *emparelhadas*, ou simplesmente *emparelhadas*, enquanto que um vértice incidente a alguma aresta M - *emparelhada* é dito M - *saturado*, e aqueles não incidentes as arestas de M são M - *expostos*.

O emparelhamento M é **maximal** se qualquer aresta não emparelhada por M possuir uma das extremidades em M . Isto é, não existe emparelhamento M' , tal que $M' \supset M$. Além disso, se M for o emparelhamento de maior cardinalidade do grafo, então M é máximo, Figura 2.4a. Sendo assim, não existe emparelhamento M' , tal que $|M'| > |M|$. Finalmente, diz-se que **emparelhamento perfeito**, Figura 2.4b, é quando todo vértice do grafo for saturado por M . Naturalmente, todo emparelhamento perfeito é máximo e todo emparelhamento máximo é maximal.

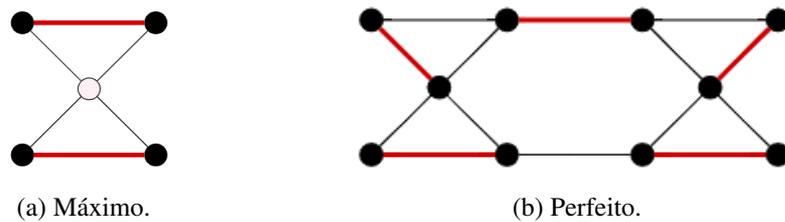


Figura 2.4. Tipos de emparelhamento: em (a) um exemplo de emparelhamento máximo e em (b) um exemplo de emparelhamento perfeito, onde todos os vértices são saturados por M .

O grafo **cobra-triangular** (*triangular snake*) - T_n , Figura 2.5, é obtido de um caminho P_n substituindo cada aresta do P_n por um ciclo C_3 [Seoud, Farid e Anwar 2022]. O grafo cobra-triangular também pode ser chamado de cacto triangular, cujo ponto de corte do bloco é um caminho. Nesse contexto, o **grafo borboleta** (*butterfly*) é o nome dado ao grafo T_3 , grafo cobra-triangular com 5 vértices.

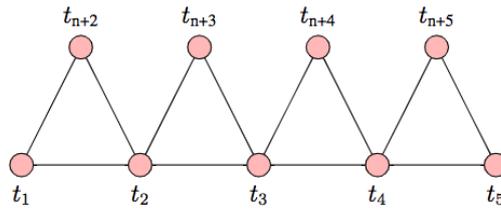


Figura 2.5. Exemplo da classe de grafo Cobra-triangular - T_5 .

Um grafo relacionado com o cobra-triangular, é o **cobra triangular alternada** (*alternate triangular snake*) - $A(T_n)$, que é obtido de um caminho u_1, u_2, \dots, u_n juntando u_i e u_{i+1} (alternadamente) a um novo vértice v_i [Jeyanthi, Maheswari e Vijayalakshmi 2019]. Ou seja, cada aresta ímpar do caminho é substituída por um ciclo com 3 vértices - C_3 , conforme apresentado na Figura 2.6.

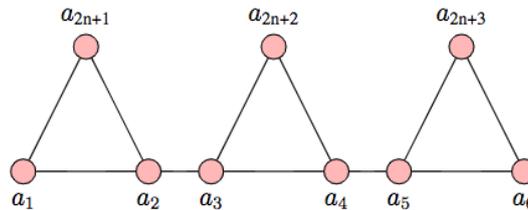


Figura 2.6. Exemplo da classe cobra-triangular alternada - A_6 .

2.3. Algoritmos e Aspectos de Complexidade Computacional

Os conceitos aqui abordados são baseados nos livros dos autores Garey [Garey e Johnson 1979] e Cormen [Cormen et al. 2012].

Um algoritmo é qualquer procedimento computacional bem definido que recebe algum valor ou conjunto de valores (instância i de P) como entrada e produz algum valor ou conjunto de valores (solução i de P) como resultado. Algoritmos diferentes criados para resolver o mesmo problema muitas vezes diferem em termos de eficiência.

A complexidade computacional estuda os recursos (tempo e espaço) necessários para executar algoritmos e resolver problemas computacionais. Estes recursos são geralmente medidos em função do tamanho da entrada do problema. A eficiência de um algoritmo depende do tamanho n da entrada. A operação básica do algoritmo é a que mais influencia no tempo de execução do algoritmo e que consome mais tempo. Portanto, analisar a eficiência de tempo de um algoritmo é contar quantas vezes a operação básica é executada para uma entrada de tamanho n .

Para a análise de eficiência de tempo de um algoritmo são usados conceitos e definições matemáticas, tal como, a análise do crescimento assintótico de funções, que tem como objetivo simplificar a análise de complexidade de tempo de execução dos algoritmos, eliminando detalhes que podem ser afetados por fatores de implementação ou hardware, avaliando como o tempo de execução do algoritmo cresce com o tamanho da entrada n , considerando um n muito grande. As principais notações são:

- O : representa o limite superior do tempo de execução ou uso de memória do algoritmo. Indica o pior caso, ou seja, determina o maior número de passos que o algoritmo pode executar;
- Ω : representa o limite inferior do tempo de execução ou uso de memória. Indica o melhor caso, ou seja, determina o menor número de passos que o algoritmo pode executar;
- Θ : representa o crescimento assintótico exato, combinando tanto o limite superior quanto o inferior. É obtido com base numa distribuição de probabilidades sobre o conjunto de entradas n .

Os algoritmos podem ser classificados de acordo com a notação O , da seguinte maneira:

- constante $O(1)$: o tempo é constante, neste caso, as instruções do algoritmo são executadas um número fixo de vezes;
- logarítmica $O(\log n)$: o tempo de execução aumenta de forma proporcional ao logaritmo do tamanho da entrada, o que significa que pode ser considerado menor do que uma constante grande;
- linear $O(n)$: o tempo de execução aumenta de forma proporcional ao tamanho da entrada. Nessa classe, normalmente, uma pequena operação é realizada em cada elemento, representando a melhor situação possível para um algoritmo que precisa processar n elementos;
- quadrática $O(n^2)$: o tempo de execução cresce proporcionalmente ao quadrado do tamanho da entrada. Algoritmos dessa classe, geralmente, surgem quando os dados são processados em pares, como ocorre em estruturas de algoritmos que envolvem um *loop* dentro de outro;
- exponencial $O(2^n)$: o tempo de execução cresce de forma exponencial com o tamanho da entrada. Essa classe de complexidade ocorre, tipicamente, em proble-

mas cuja solução exige a aplicação de força bruta, explorando todas as possibilidades para encontrar a resposta.

Essa classificação auxilia na compreensão do comportamento de um algoritmo à medida que o tamanho da entrada cresce, permitindo compará-los teoricamente e ajudando na seleção de algoritmos apropriados para resolver problemas específicos.

Capítulo 3

O Jogo da Torre de Hanoi

Este capítulo apresenta os aspectos matemáticos e computacionais da Torre de Hanoi, sua estrutura e regras. Em seguida, são apresentados algoritmos para resolução do jogo em diferentes cenários. Por fim, apresenta-se o grafo do jogo baseado no Triângulo de Sierpinski, seguido de algumas aplicações da ToH.

3.1. O Jogo, Estrutura e Regras

A Torre de Hanoi foi inventada pelo matemático francês Edouard Lucas em 1883 [Al-lardice e Fraser 1883]. Sua origem foi inspirada em uma lenda hindu [Hofstadter 1983], apresentada na Figura 3.1, que fala de um templo em Benares, cidade sagrada da Índia.

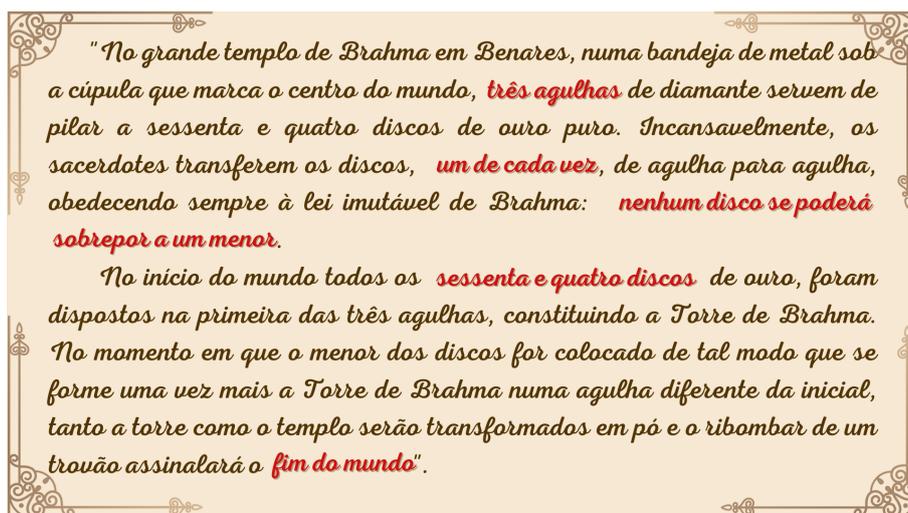


Figura 3.1. Lenda que inspirou a estrutura e regras do jogo da Torre de Hanoi.

Sobre o nome Torre de Hanoi, pensa-se logo na capital do Vietnã, Hanoi. No entanto, quando Lucas começou a comercializar o jogo com apenas 8 discos de madeira,

os jornais estavam cheios de relatórios de Tonkin. Na verdade, Hanoi havia sido tomada em 1882, mas durante o verão de 1883 estava sob cerco constante por tropas da província chinesa de Yunnan. Mas, em 25 de agosto desse mesmo ano, o tratado de Harmand estabeleceu o governo da França sobre Annam e Tonkin. Adicionalmente, ainda em 1883, Parville publica uma variação da ToH, onde discos de diâmetros crescentes foram substituídos por pirâmides ocas de tamanho decrescente, e a chama de “questão de Tonkin” e comenta o fato de que os discos da ToH de Lucas eram de madeira em vez de ouro, como sendo mais prudente porque diz respeito a Tonkin. Então, Lucas selecionou o nome Hanoi, porque estava nas manchetes na época [Parville 1884] apud [Hinz, Klavžar e Petr 2018].

A Torre de Hanoi possui uma base com o conjunto de pinos $P = \{p_0, p_1, \dots, p_k\}$, para algum natural $k \geq 3$, e o conjunto de discos $D = \{d_1, d_2, \dots, d_n\}$, onde $n \in \mathbb{N}$, como apresentado na Figura 3.2a. Neste trabalho, considera-se apenas a versão da ToH tradicional com três pinos verticais, ou seja, $k = 3$.

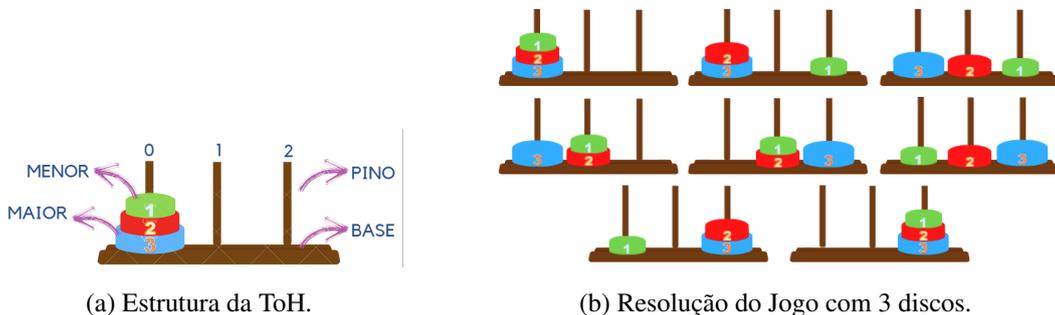


Figura 3.2. Torre de Hanoi: (a) representação tangível do jogo com 3 pinos, com o menor disco, na cor verde, no topo da pilha e o maior disco, na cor azul, na base; (b) demonstração da solução do jogo com 3 discos, onde os discos são movidos de p_0 para p_2 , até que o objetivo do jogo seja alcançado.

As configurações do jogo da ToH podem ser classificadas em três tipos de estados, adaptados de Hinz et al. [2018]: regular, perfeito e irregular, que serão explicitados nas Definições 1, 2 e 3, respectivamente.

Definição 1 (Estado regular). *Um estado **regular** é aquele em que a distribuição dos n discos, nos 3 pinos, respeita as regras do jogo.*

Definição 2 (Estado perfeito). *Um estado **perfeito** é um estado regular em que todos os discos se encontram em um único pino.*

Definição 3 (Estado irregular). *Um estado **irregular** é uma distribuição de discos entre os pinos, em que, pelo menos uma regra do jogo é violada.*

No estado perfeito inicial do jogo, todos os discos estão empilhados no pino p_0 na seguinte ordem: d_1, d_2, \dots, d_n , sendo que d_1 está no topo da pilha. O objetivo do jogo é alcançar o estado perfeito final, ou seja, colocar todos os discos em p_2 nessa mesma disposição, obedecendo as seguintes regras [Er 1983]:

- i. em cada movimento, somente o topo de um dos pinos pode ser movido;
- ii. o disco d_i só pode ser colocado sobre d_j , se $i < j$.

O número mínimo de movimentos para solucionar o jogo da ToH depende do número de discos. A Figura 3.2b tem a ilustração e resolução do jogo com 3 discos, usando o menor número de movimentos. Na Tabela 3.1, pode-se acompanhar o número mínimo de movimentos para o jogo com 1, 2, 3, 4, ..., n discos. Assim, observando a generalização matemática, é possível notar que, seguindo as regras descritas anteriormente, o número mínimo de passos para mover todos os discos do estado perfeito inicial ao estado perfeito final do jogo é $2^n - 1$, em que n é o número de discos.

Tabela 3.1. Torre de Hanoi: cálculo matemático para encontrar o número mínimo de movimentos com 1, 2, 3, 4, 5, ..., n discos.

Nº de Discos	Nº Mínimo de Movimentos	Generalização Matemática
1	1	$2^1 - 1 = 1$
2	3	$2^2 - 1 = 3$
3	7	$2^3 - 1 = 7$
4	15	$2^4 - 1 = 15$
5	31	$2^5 - 1 = 31$
...
n	x	$2^n - 1 = x$

Para 64 discos, como na descrição original do problema apresentado na Figura 3.1, são necessários 18.446.744.073.709.551.615 movimentos. A menor sequência de movimentos que leva do estado perfeito inicial ao estado perfeito final do jogo é chamada de caminho ótimo [Romik 2006]. A tarefa da Torre de Hanoi Clássica para n discos tem uma única solução ótima [Scorer, Grundy e Smith 1944].

Para solucionar o jogo com o menor número de movimentos move-se o disco maior (d_n) apenas uma vez. Então, isso se dá da seguinte maneira: os $n - 1$, primeiros

discos, serão movidos de p_0 para p_1 ; em seguida, move-se o disco d_n de p_0 para p_2 e depois os $n - 1$ discos serão movidos de p_1 para p_2 .

3.2. Aspectos Algorítmicos

O algoritmo para resolver a ToH pode ser recursivo, como exibido no Algoritmo [1](#) ou iterativo, como o exposto no Algoritmo [2](#). O Algoritmo [1](#) realiza duas chamadas recursivas e um movimento no caso geral e um único movimento no caso base, o que pode ser expresso pela seguinte recorrência:

Algoritmo 1: Torre de Hanoi - Algoritmo Recursivo.	
	Entrada: d_1, d_2, \dots, d_n e p_0, p_1, p_2 de mesma capacidade. Estado inicial: todos os n discos empilhados em p_0 .
	Saída: Estado final: todos os discos empilhados em p_2 .
1	Hanoi (n, p_0, p_2, p_1)
2	se ($n = 1$) então
3	MoveDisco($1, p_0, p_2$)
4	senão
5	Hanoi ($n - 1, p_0, p_1, p_2$)
6	MoveDisco(n, p_0, p_2)
7	Hanoi ($n - 1, p_1, p_2, p_0$)
8	fim
9	fim

$$T(n) = \begin{cases} 1, & \text{se } n=1 \\ 2T(n-1) + 1, & \text{se } n>1 \end{cases}$$

Resolvendo a equação de recorrência, conclui-se que a função de custo do Algoritmo [1](#) é $T(n) = 2^n - 1$, para $n \geq 1$, do qual obtêm-se a complexidade de tempo $O(2^n)$.

O algoritmo iterativo, exibido no Algoritmo [2](#), segue uma **direção certa** para mover os discos de acordo com a paridade do número de discos n . Essa direção pode ser entendida como uma lista circular, em que o último elemento está logicamente conectado com o primeiro, formando um ciclo. Assim, se o número de discos n é par a direção certa do d_1 é no sentido anti-horário, enquanto, se n for ímpar será no sentido horário. Por exemplo, se n for par e d_1 está em p_0 , ele será movido na sequência: $p_1, p_2, p_0, \dots, p_2$ e se n é ímpar, a sequência será: $p_2, p_1, p_0, \dots, p_2$ [\[Er 1986\]](#). É conhecido que o algoritmo

iterativo da ToH (Algoritmo 2) soluciona o jogo com o número mínimo de movimentos, isto é, resultando na complexidade $O(2^n)$ [Hinz, Klavžar e Petr 2018].

Algoritmo 2: Torre de Hanoi - Algoritmo Iterativo.	
Entrada: d_1, d_2, \dots, d_n e p_0, p_1, p_2 de mesma capacidade. Estado inicial: todos os discos empilhados em p_0 (origem).	
Saída: Estado final: todos os discos empilhados em p_2 (destino).	
1	início
2	repita
3	mover d_1 uma posição à frente na direção certa
4	realizar o único movimento possível
5	até todos os discos estejam em p_2;
6	fim

Assim, os algoritmos recursivo e iterativo solucionam o jogo da Torre de Hanoi levando os discos de p_0 para p_2 . Porém, se alguém iniciar o jogo e por algum motivo precisar parar, o jogo estará em um estado regular s , para continuá-lo apresenta-se no Algoritmo 3 que determina qual o primeiro movimento necessário para continuar solucionando o jogo com o menor número de movimentos possíveis [Hinz, Klavžar e Petr 2018]. Dessa forma, dado que s é um estado regular que se encontra no caminho ótimo e d_i é o menor disco no topo de um dos pinos, sendo $d_i \neq d_1$, o jogador tem que decidir mover d_1 ou d_i . A direção certa de d_i – também se dará de maneira cíclica – na direção oposta a d_1 se i for par e na mesma direção de d_1 , caso contrário. Então, o Algoritmo 3 primeiro verifica se mover d_i na direção certa é um movimento legal. Em caso afirmativo, significa que o jogo foi abandonado após um movimento de d_1 .

Após executar o primeiro movimento ótimo de um estado abandonado, Algoritmo 3 chama-se o algoritmo iterativo e prossegue com a resolução do jogo até que todos os discos estejam no pino de destino, ou seja, em p_2 .

Outra situação que pode ocorrer, é por exemplo, uma pessoa estar brincando e não conhecer nenhum algoritmo de solução do jogo. Nesse caso, é muito provável que o estado em que o jogo se encontra, caso as regras sejam seguidas, seja um estado regular fora do caminho ótimo. Então, Hinz et al. [2018] tem um algoritmo baseado em um autômato que detecta se o jogo se encontra no caminho ótimo, baseando-se neste algoritmo, propõe-se o Algoritmo 4 para detectar se o estado regular, em que o jogo se encontra, está

Algoritmo 3: ToH - 1º Movimento Ótimo de um Estado Abandonado.

Entrada: d_1, d_2, \dots, d_n e p_0, p_1, p_2 de mesma capacidade, s estado regular no caminho ótimo.

```
1 início
2   seja  $d_i$  o menor disco superior de  $s$  diferente de  $d_1$ 
3   se o movimento legal de  $d_i$  está na direção certa então
4     mover disco  $d_i$  na direção certa
5   senão
6     mover o disco  $d_1$  na direção certa
7   fim
8 fim
```

no caminho ótimo para solucionar o problema.

Como no jogo clássico da Torre de Hanoi só existe uma única solução ótima [Scorer, Grundy e Smith 1944], ou seja, só existe um caminho ótimo, então, propõe-se o Algoritmo 4 para descobrir se o jogo da ToH está no caminho ótimo. Inicialmente o Algoritmo 4 verifica se d_n está em p_1 . Pois, na solução ótima do jogo d_n se move exatamente uma vez, ou seja, de p_0 para p_2 . Então, d_n tem que estar em p_0 ou p_2 , qualquer estado que d_n esteja em p_1 , este não está no caminho ótimo. Logo, o algoritmo retorna "NÃO". Se em s , d_n está no pino inicial, d_{n-1} também deve estar no pino inicial ou no pino auxiliar. Da mesma forma, se d_n estiver no destino, d_{n-1} também deve estar no destino ou no auxiliar. Nesse caso, o algoritmo retorna "SIM". Caso contrário, deve informar que s "NÃO" está no caminho ótimo. Dessa forma, o Algoritmo 4 faz no máximo uma chamada recursiva, ou seja, esse algoritmo é linear no número de discos.

3.3. Formas de Representação de um Estado do Jogo

Na literatura são encontradas algumas rotulações de um estado do jogo da ToH, como por exemplo, a Representação Binária (Apêndice A) e o Código Gray (Apêndice B). Porém, nesta dissertação, para melhor adequação as restrições de todas as variações, optou-se por utilizar a nomenclatura abordada na Seção 3.3.1.

3.3.1. Rótulo de um Estado

Os rótulos de H_p^n foram definidos como sendo sequências de números tal que:

- o primeiro número indica o disco que está no topo da pilha;

Algoritmo 4: ToH - Detecção de Desvio do Caminho Ótimo.

Entrada: d_1, d_2, \dots, d_n ; 3 pinos de mesma capacidade: p_0, p_1 e p_2 ; e s um estado regular do jogo.

Saída: "SIM", está no caminho ótimo ou "NAO", não está no caminho ótimo.

```

1 ehOtimo ( $p_0, p_1, p_2, n$ )
2   se ( $d_n \in p_1$ ) então
3     retorna "NÃO"
4   senão
5     se ( $n = 1$ ) então
6       retorna "SIM"
7     senão
8       se ( $d_n \in p_0$ ) então
9         retorna ehOtimo ( $p_0, p_2, p_1, n - 1$ )
10      senão
11        retorna ehOtimo ( $p_1, p_0, p_2, n - 1$ )
12      fim
13    fim
14  fim
15 fim

```

- a “/” (barra) indica que os números seguintes estão no próximo pino/pilha;
- e assim, sucessivamente.

Por exemplo, o rótulo “12/3/” indica que d_1 e d_2 estão em p_0 e d_3 está em p_1 . Assim como “1//23”, demonstra que o disco d_1 está em p_0 e os discos d_2 e d_3 estão no p_2 .

3.4. Grafo da Torre de Hanoi Baseado no Triângulo de Sierpinski

O grafo H_p^n será apresentado nesta seção como é tradicionalmente visto na literatura, ou seja, definido com base no Triângulo de Sierpinski [Scorer, Grundy e Smith 1944] e na Seção 6.1 será proposta uma definição recursiva do grafo da Torre de Hanoi que se restringe ao domínio da Teoria dos Grafos.

O grafo da ToH caracteriza todos os estados regulares possíveis no jogo, bem como os movimentos que podem ser feitos de um estado para outro [Zhang et al. 2016]. Denota-se H_p^n o grafo da ToH com n discos e p pinos. O conjunto de vértices de H_p^n são os estados regulares do jogo e as arestas são as transições entre esses estados. Adicional-

mente, H_p^n tem p^n vértices e $\frac{1}{2} \binom{p}{2} (p^n - (p-2)^n)$ arestas [Eppstein, Frishberg e Maxwell 2022]. Dois vértices (isto é, estados regulares) são adjacentes se forem obtidos um do outro por um movimento legal de um disco.

O Triângulo de Sierpinski¹ é uma figura geométrica fractal, estudada pelo matemático polonês Waclav Sierpinski, em 1915 [Sierpinski 1915], cuja construção é realizada por um processo iterativo de divisão de triângulos. É uma das formas elementares da geometria fractal por apresentar algumas propriedades [Hinz, Klavžar e Petr 2018]:

- ter tantos pontos como o do conjunto dos números reais;
- ter área igual a zero;
- ser autossemelhante (cada parte sua é idêntica ao todo);
- não perder a sua definição inicial à medida que é ampliado.

Os estados do jogo da ToH foram nomeados, conforme o rótulo abordado na Seção 3.3.1. O Grafo, apresentado na Figura 3.5b, permite observar que a medida que o número de discos aumenta, o grafo se torna cada vez mais parecido com o grafo do Triângulo de Sierpinski. Sendo que, há uma aresta entre cada H_3^1 , onde esse espaço é chamado de α e que se tender α a zero, o grafo H_3^n tende ao Triângulo de Sierpinski.

A **correspondência de Lucas** afirma que a medida que n se torna grande, H_3^n se torna um fractal com uma notável semelhança com a estrutura de Sierpinski e o triângulo de Pascal de módulo 2, todos com dimensão fractal $\log_2 3$ [Poole 1994].

Para caracterizar H_3^n , Poole [1994] propõe a Proposição 1.

Proposição 1 (Propriedades do grafo H_3^n). *Para todos os inteiros positivos n , H_3^n é um grafo planar, biconexo, hamiltoniano de ordem 3^n , de diâmetro de $2^n - 1$, com exatamente 3 vértices de grau 2 e todos os outros vértices de grau 3 [Poole 1994].*

Para o jogo da ToH com apenas 1 disco têm-se 3 estados possíveis, como mostrado na Figura 3.3a e no grafo da Figura 3.3b, o disco está em p_0 , o disco está em p_1 ou o disco está em p_2 .

Na ToH com 2 discos, percebe-se que cada posição no jogo com 2 discos corresponde a uma sequência de dois algarismos, como na Figura 3.4a. H_3^2 tem $3^2 = 9$ vértices.

¹O processo de construção do Triângulo de Sierpinski e mais informações sobre ele, encontram-se no Apêndice C.

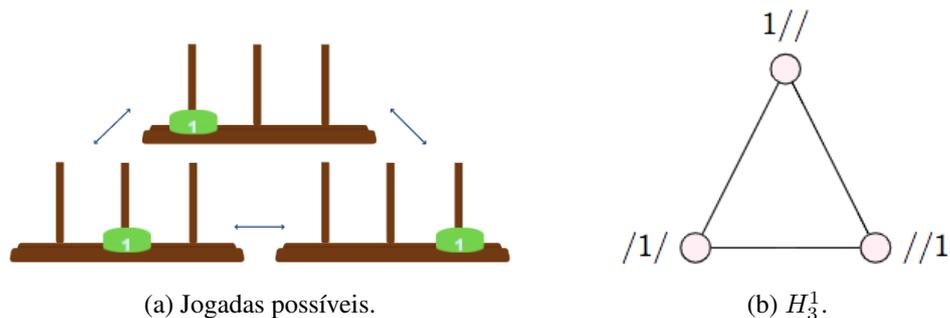


Figura 3.3. Torre de Hanoi com 1 disco: (a) estados possíveis do jogo; (b) a representação gráfica desses pontos (estados) no grafo, formando assim, o grafo H_3^1 .

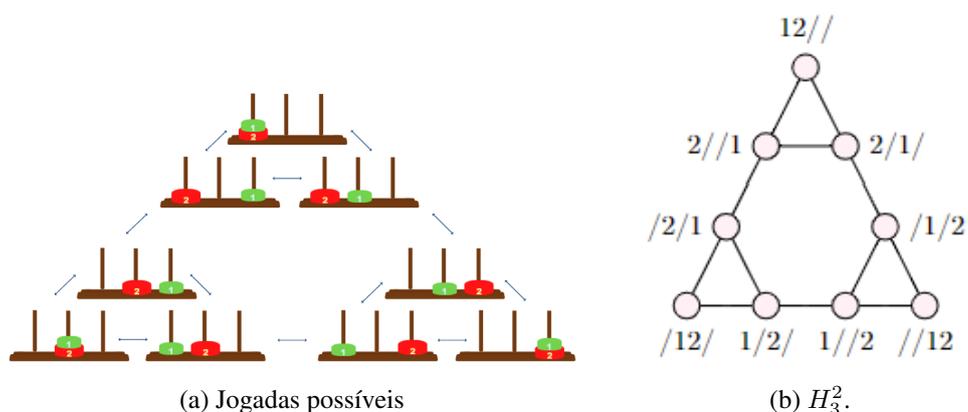


Figura 3.4. Torre de Hanoi com 2 discos: em (a) todos os estados possíveis do jogo e em (b) a representação gráfica do grafo H_3^2 .

Observa-se que, como os movimentos possíveis no jogo da ToH são reversíveis, os grafos da torre não são direcionados. Além disso, como é “legal” caminhar de qualquer vértice para qualquer outro vértice usando arestas adjacentes, o grafo H_3^2 , exibido na Figura 3.4b, é conexo. E como não há cruzamento de arestas, o grafo também é planar.

Para solucionar o jogo com o menor número de movimentos possível, deve-se pegar o caminho para a direita do grafo H_3^2 . O comprimento desse caminho corresponde a 3 arestas, o que torna consistente com o que já foi demonstrado na Tabela 3.1. O grafo H_3^2 é formado por um caso especial da família cobra triangular, o cobra triangular alternada - $A(T_6)$.

Sendo assim, obedecendo as regras do jogo com 3 discos, pode-se montar o grafo H_3^3 com 27 vértices. Portanto, H_3^3 fica como na Figura 3.5b, ao observar este grafo, percebe-se que ele é formado por três cópias de um grafo menor (de fato, H_3^2 - Figura 3.4b), ligadas por três arestas. Similarmente, o grafo H_3^2 - Figura 3.4b, também é formado

por três cópias de H_3^1 - Figura 3.3b. Por fim, H_3^3 possui um caminho de qualquer vértice para qualquer outro, como afirma o Lema 1.

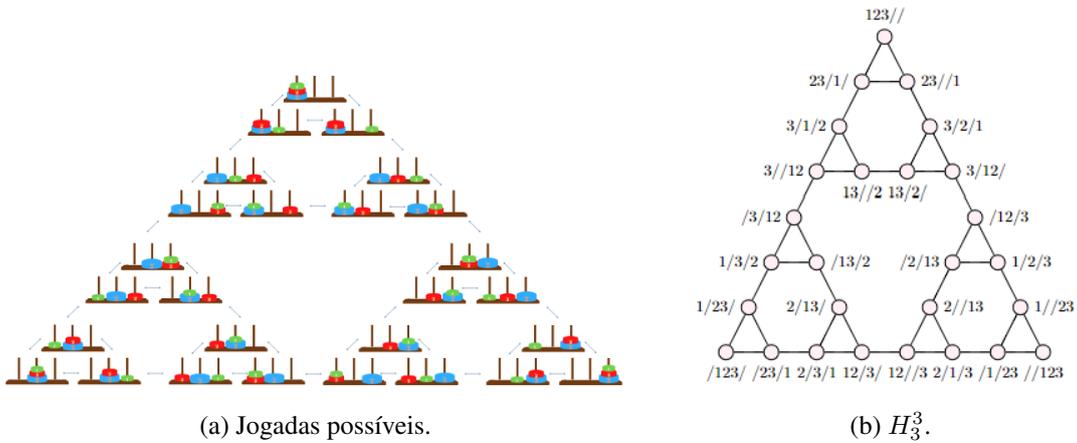


Figura 3.5. Torre de Hanoi com 3 discos: em (a) todos os estados possíveis do jogo e em (b) a representação gráfica do grafo H_3^3 .

Lema 1 (Caminho em H_3^n). Para todos os inteiros positivos n , existe um caminho em H_3^n entre qualquer par de vértices distintos [Scorer, Grundy e Smith 1944].

Em 1944, Scorer, Grundy e Smith provaram o Lema 1 que, independente do valor de n , sempre existirá um caminho entre qualquer par de vértices. Logo, H_3^n é um grafo conexo. Assim, o caminho mínimo do estado inicial para o estado final corre em linha reta ao longo de uma das margens do grafo (123//, 23//1, 3/2/1, 3/12/, /12/3, 1/2/3, 1//3, //123), tendo a dimensão de $2^n - 1$, como apresentado na Figura 3.6.

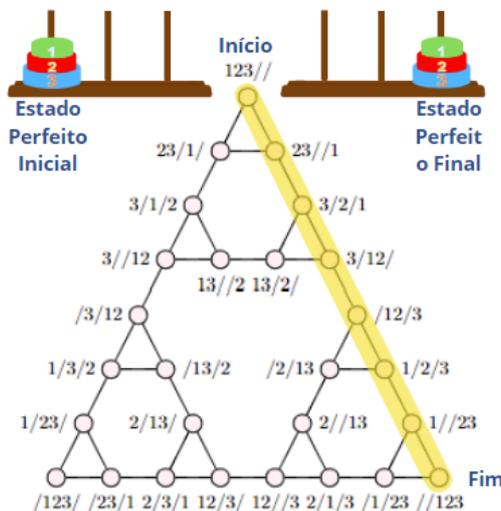


Figura 3.6. H_3^3 : Caminho do exemplo da solução do jogo, com o menor número de movimentos, no grafo.

Portanto, analisando as Proposições dispostas no estado da arte, como apresentado na Proposição [1], chega-se a conclusão de que H_3^n , é um grafo hamiltoniano, simples, não direcionado, biconexo, de ordem 3^n , de diâmetro de $2^n - 1$, com exatamente 3 vértices de grau 2 e todos os outros vértices de grau 3, baseado no triângulo de Sierpinski, para todos os inteiros positivos $n > 2$.

3.5. Aplicações da Torre de Hanoi

Há diversas aplicações do jogo da ToH indicadas na literatura [Martins et al. 2022], sendo utilizadas em processos de ensino-aprendizagem de conteúdos de matemática, como por exemplo, nos conteúdos de função exponencial [Oliveira e Calejon 2016], potenciação [Oliveira, Brim e Pinheiro 2019], progressão geométrica, geometria plana e descritiva. Em computação, é usada para aprendizagem de forma lúdica de algoritmos recursivos [Santiago e Kronbauer 2016] e iterativos [Andrade et al. 2016], com suas estruturas de dados em vetores, pilhas e listas circulares.

Ainda em computação, pode-se explorar aspectos de complexidade computacional de problemas, em que o jogo da ToH se configura como sendo exponencial, pois são necessários $O(2^n)$ movimentos para solucioná-lo. No cenário atual, a ToH tem sido muito utilizada no desenvolvimento da lógica [Pinho et al. 2016] e da disseminação do pensamento computacional [Silva, Limberger e Pellanda 2014] [Freitas et al. 2019].

Na indústria, problemas baseados em torres ou problemas de empilhamento (do inglês, *pile problems*), são encontrados, principalmente, em processos logísticos, como no armazenamento e no transporte de produtos, nos quais a carga tem restrição quanto à altura do empilhamento, prioridade de descarregamento, diferença de tamanho/peso ou, ainda, a fragilidade da mercadoria [Hempel, Schmiedel e Kämmerer 2006]. Nesses problemas, deve-se encontrar soluções para realizar o transporte da carga de forma a satisfazer as condições estabelecidas previamente e, principalmente, maximar a utilização do espaço para a realização do transporte adequado [Huang e Lin 2012], tais como nas cargas e descargas de caminhões, navios ou trens [Schulz 1989] e no armazenamento de produtos, sendo, geralmente, problemas de otimização combinatória NP-difíceis [Kämmerer e Hempel 1997].

Ademais, têm-se, ainda, várias presenças da Torre de Hanoi na Cultura Popular, onde podem-se apontar vários filmes, tais como: “Now Inhale” (“Agora Inspire”) de

Eric Frank Russell, "The Celestial Toymaker"("O fabricante de brinquedos celestial") de Doctor Who em 1966, "A Ascensão do Planeta dos Macacos"de 2011 e em "Star Wars: Knights of the Old Republic". O objetivo da empregabilidade do jogo, nesses filmes, foi incentivar o pensamento computacional e/ou o raciocínio lógico dos personagens na história.

Portanto, muitas situações na vida real apresentam uma configuração de itens empilhados, formando uma torre, que precisam ser transferidos para outra, com a ordem dos itens diferentes e prioridade de tamanho e/ou pesos [Martins et al. 2022]. O Apêndice **D** aborda esses problemas e o Apêndice **E** mostra um exemplo de aplicação na área de Tecnologia da Informação (TI).

Capítulo 4

Outros Jogos e Testes Baseados em Torres

Este capítulo traz os principais trabalhos encontrados na literatura, relacionados aos jogos de torres que serão explorados nas seções posteriores. Nesse sentido, para cada um desses jogos, são explicadas as regras, objetivos e modelos em grafos.

4.1. Classificação por Características Estruturais e Jogabilidade

De acordo com a classificação das variações do jogo da Torre de Hanoi explorada através da Tabela 4.1, a complexidade computacional permite comparações com os trabalhos correlacionados no estudo teórico envolvido com as Torres de Hanoi e as demais variações, devido às aparentes semelhanças em suas estruturas e demanda de soluções.

Foram encontradas algumas correlações significativas, no que diz respeito à estrutura, como: o número de pinos, a capacidade dos pinos, as regras do jogo, os algoritmos, a complexidade computacional e o modelo em grafos.

Desse modo, para considerar um jogo como uma variante da Torre de Hanoi, definiu-se a classificação da seguinte forma: estrutura, regras do jogo e aspectos teóricos, a partir da análise das características pontuadas abaixo:

- i. número de pinos;
- ii. capacidade dos pinos;
- iii. número de discos;
- iv. discos de tamanho/cor diferente;

Tabela 4.1. Principais resultados na literatura.

Quanto a	Características	ToH	ToB	ToO	ToL	Cíclica	Reve	Multipino
Estrutura	Número de pinos	3	3	3	3	h	4	>3
	Capacidade dos pinos	n	n	n	n	3, 2, 1	n	n
	Número de discos	n	n	n	n	3	n	n
	Discos de tamanho/cor diferentes	tam	tam	cor	cor	tam	tam	tam
Regras	Move 1 disco por vez	sim	sim	sim	sim	sim	sim	sim
	Disco maior NÃO pode em cima do menor	sim	sim	-	sim	-	sim	sim
	Permite discos adjacentes de mesma cor	sim	sim	sim	sim	não	sim	sim
	Só move para pino adjacentes	não	sim	não	não	não	não	não
Aspectos Teóricos	Algoritmo	rec/ite	rec	ite	-	ite	ite	ite
	Complexidade do Problema	2^{n-1}	3^{n-1}	-	-	$\sqrt[n]{h \log h + 2h}$	$\sqrt{\pi} 2^{\sqrt{2n}}$	2^{n-1}
Referências	Modelo em Grafos	sim	sim	sim	sim	não	não	não
	-	[Poole 1994]	[Herter e Rote 2018]	[Unterrainer et al. 2005]	[Hinz et al. 2009]	[Berend e Sapir 2006]	[Er 1988]	[Brandstädt, Le e Spinrad 1999]

- v. se move um disco por vez;
- vi. se o disco maior não pode ficar em cima do menor;
- vii. se permite discos adjacentes de mesma cor;
- viii. se só é possível mover os discos para pinos adjacentes;
- ix. se possui algoritmo;
- x. qual a complexidade do problema;
- xi. se possui modelo em grafos.

Embora as convenções acima pareçam bastante restritivas, elas oferecem um grande número de variações, por exemplo: pode haver um número arbitrário de pinos; os pinos também podem ser distinguidos por suas capacidades, isto é, pelo número de discos que eles podem sustentar; os discos podem ser diferenciados em tamanho e/ou cor; mais de um disco superior pode ser movido em um único movimento; pode haver restrições ou relaxamentos adicionais nos movimentos; e qualquer combinação dos anteriores.

4.2. Torre de Bucarest

A Torre de Bucarest (ToB) dispõe de uma estrutura igual a da Torre de Hanoi, como pode ser visto na Figura 4.1a, mas possui uma regra de movimento [Herter e Rote 2018], como é especificado na Proposição 2.

Proposição 2 (Regra de movimento da ToB). *Descrição dos movimentos na ToB:*

1. Na Torre de Hanoi, existem três movimentos possíveis a partir de qualquer posição, exceto quando todos os discos estão no mesmo pino: nesses casos, há apenas dois movimentos possíveis.
2. Na Torre de Bucarest, existem dois movimentos possíveis a partir de qualquer posição, exceto quando todos os discos estão em p_0 ou p_2 : nesses casos, há apenas um movimento possível.

Como visto no Capítulo 3, o Algoritmo 2 para resolver o jogo da ToH segue o sentido horário ou anti-horário. Já a Torre de Bucarest não admite solução nesse formato circular, pois a Torre de Bucarest consiste em três regras básicas:

- i. em cada movimento, somente o topo de um dos pinos pode ser movido;
- ii. o disco d_i só pode ser colocado sobre d_j , se $i < j$;
- iii. um disco só pode ser movido para um pino adjacente, ou seja, de p_0 não se pode mover o disco para p_2 , nem de p_2 para p_0 .

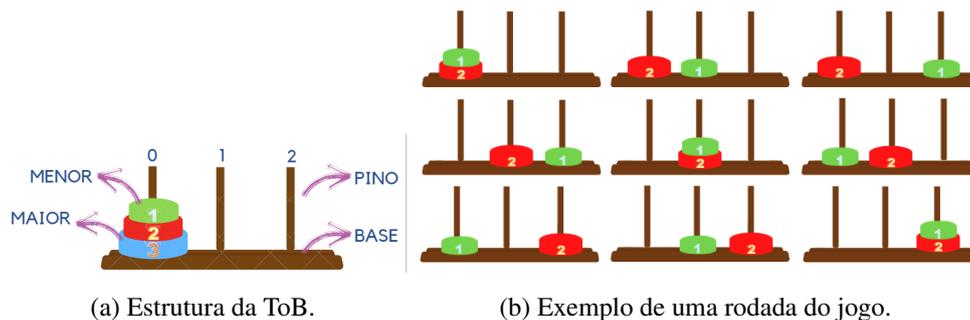


Figura 4.1. Torre de Bucarest: (a) representação tangível do jogo com 3 pinos, com o menor disco, na cor verde, no topo da pilha e o maior disco, na cor azul, na base; (b) demonstração da solução do jogo com 3 discos, onde os discos são movidos de p_0 para p_2 obedecendo as regras, até que o objetivo do jogo seja alcançado.

Como na ToH, o número de movimentos, na ToB, depende do número de discos. Na Figura 4.1b tem-se a demonstração do funcionamento do jogo com 2 discos. Na Tabela 4.2, pode-se acompanhar o número mínimo de movimentos para o jogo com 1, 2, 3, 4, ..., n discos. Observando a generalização matemática é possível ver que o número mínimo de passos para mover todos os discos de p_0 para p_2 é $3^n - 1$, onde n é o número de discos.

Tabela 4.2. Torre de Bucarest: cálculo matemático para encontrar o número mínimo de movimentos com 1, 2, 3, 4, 5, ..., n discos.

Nº de Discos	Nº Mínimo de Movimentos	Generalização Matemática
1	2	$3^1 - 1 = 2$
2	8	$3^2 - 1 = 8$
3	26	$3^3 - 1 = 26$
4	80	$3^4 - 1 = 80$
5	242	$3^5 - 1 = 242$
...
n	x	$3^n - 1 = x$

4.2.1. Aspectos Algorítmicos

Aplicando-se a regra em que o movimento de disco só pode ser feito entre pinos adjacentes, o algoritmo iterativo da resolução da ToB, como exibido no Algoritmo 5, é similar ao Algoritmo 2, mas sem a necessidade de tratar o caso do número de discos ser par ou ímpar, isto simplifica o algoritmo.

O jogo da ToB também pode ser solucionado recursivamente com o Algoritmo 6.

Algoritmo 5: ToB - Algoritmo Iterativo.

Entrada: n discos de tamanhos diferentes; e 3 pinos (p_0 (origem), p_1 (auxiliar), p_2 (destino)) de mesma capacidade. Estado inicial: todos os n discos empilhados em p_0 (origem).

Saída: Estado final: todos os discos empilhados em p_2 (destino).

```

1 início
2   repita
3     mover  $d_1$  2 posições possíveis (horário ou anti-horário)
4     realizar o próximo movimento possível
5   até todos os discos empilhados em  $p_2$ ;
6 fim

```

Algoritmo 6: Torre de Bucareste - Algoritmo Recursivo.

Entrada: n discos de tamanhos diferentes; e 3 pinos (p_0 (origem), p_1 (auxiliar), p_2 (destino)) de mesma capacidade. Estado inicial: todos os n discos empilhados em p_0 (origem).

Saída: Estado final: todos os discos empilhados em p_2 (destino).

```

1 Bucareste ( $n, p_0, p_1, p_2$ )
2   se ( $n > 0$ ) então
3     Bucareste ( $n - 1, p_0, p_2, p_1$ )
4     Bucareste ( $n - 1, p_1, p_0, p_2$ )
5     mover o disco do topo de  $p_0$  para o topo de  $p_1$ 
6     Bucareste ( $n - 1, p_2, p_0, p_1$ )
7     Bucareste ( $n - 1, p_1, p_0, p_2$ )
8     mover o disco do topo da  $p_1$  para o topo de  $p_2$ 
9     Bucareste ( $n - 1, p_0, p_2, p_1$ )
10    Bucareste ( $n - 1, p_1, p_2, p_0$ )
11  fim
12 fim

```

São necessários $3^n - 1$ movimentos para solucionar o jogo da ToB [Herter e Rote 2018]. Sendo assim, a complexidade de tempo é $O(3^n)$. Portanto, o algoritmo desse jogo é exponencial, pois o número de movimentos para solucionar o jogo é exponencial.

4.2.2. Modelagem em Grafos

O jogo da ToB também pode ser associado a um grafo $B_p^n = (V, E)$, no qual n é o número de discos e p equivale ao número de pinos no jogo. O conjunto de vértices representa os

estados regulares no jogo e E as arestas que correspondem aos movimentos permitidos entre os estados regulares do jogo. Os estados do jogo são nomeados conforme a Seção 3.3.1 e as propriedades do grafo desse jogo são expressas na Proposição 3.

Proposição 3 (Propriedades do grafo B_3^n). *É um grafo simples, não direcionado, conexo, planar, tem sempre 2 vértices de grau 1 e os demais são de grau 2, grau máximo 2 e é um grafo do tipo caminho.*

Para embasar a Proposição 3, inicia-se com a ToB com 1 disco - B_3^1 , apresentado na Figura 4.2a, tem 2 vértices de grau 1 e 1 vértice de grau 2, o que o torna um caminho P_3 . Adicionando mais 1 disco, no caso, o grafo B_3^2 , o que pode ser observado na Figura 4.2b, o grafo passa a ter 9 vértices, continuando 2 de grau 1, mas agora com 7 vértices de grau 2.

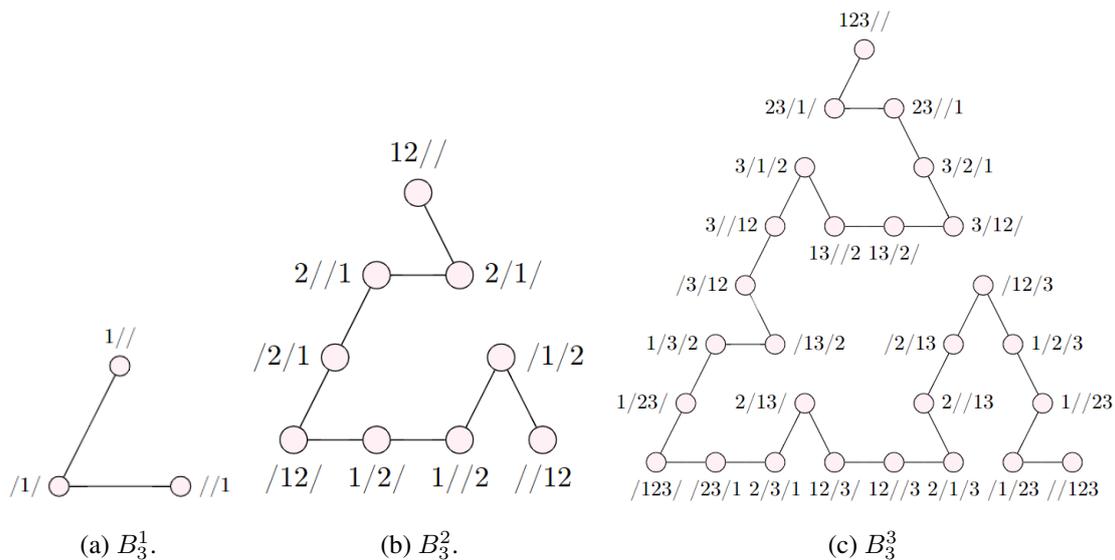


Figura 4.2. Grafos da Torre de Bucareste.

Analisando o grafo B_3^3 , exposto na Figura 4.2c, este grafo possui 27 vértices e 26 arestas, que formam um caminho, onde os discos são levados do p_0 para o p_2 , dispõe de 25 vértices de grau 2 e somente 2 vértices de grau 1, sendo portanto, de grau máximo 2.

A partir desse contexto, B_p^n possui algumas características tais quais ao grafo da ToH - H_p^n , dentre elas, a principal é o número de vértices: 3^n , que corresponde as formas possíveis de atribuir os discos aos pinos p_0, p_1 e p_2 , ou seja, o número de estados regulares possíveis no jogo.

Um exemplo do jogo da ToB com 3 discos é demonstrado na Figura 4.3, no qual

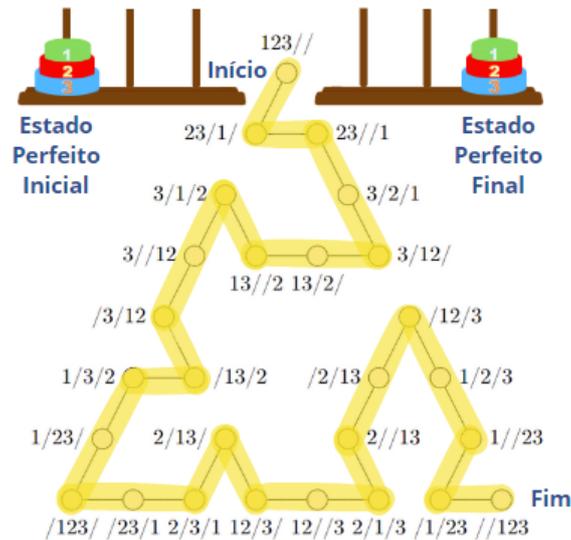


Figura 4.3. Grafo B_3^3 com exemplo de uma jogada com 3 discos, em amarelo o caminho percorrido do estado inicial até o estado final do jogo.

se pode observar o estado perfeito inicial e o estado perfeito final. Logo, fica visível o Caminho Hamiltoniano, em amarelo, formado pelos movimentos para solucionar o jogo, pois, parte-se do estado inicial perfeito e passa por todos os vértices de B_3^3 .

4.3. Torre de Londres

Em seu estudo, de 1982, sobre os défices de planejamentos exibidos por pacientes com lesão do lobo frontal, Shallice propôs uma variação da clássica Torre de Hanoi e como um nativo da Inglaterra, a apelidou de Torre de Londres (ToL) [Berg e Byrd 2002]. Esse jogo é amplamente utilizado para avaliar planejamento e resolução de problemas [Such et al. 2011].

A ToL, como apresentada na Figura 4.4a possui 3 esferas de mesmo tamanho e cores diferentes, uma base retangular contendo 3 pinos de tamanhos diferentes: p_0 , p_1 e p_2 , sendo que o mais curto sustenta apenas 1 esfera e o mais longo 3 esferas [Shallice 1982]. O objetivo do jogo é mover as esferas da posição inicial, Figura 4.4a, até chegar à configuração do cartão-problema, Figura 4.4b, obedecendo sempre as duas regras do jogo:

- i. mover uma esfera por vez;
- ii. uma esfera só pode ser movida para um pino se ele não tiver atingido a sua capacidade máxima. A capacidade de $p_0 = n$, $p_1 = n - 1$ e $p_2 = n - 2$ esferas.

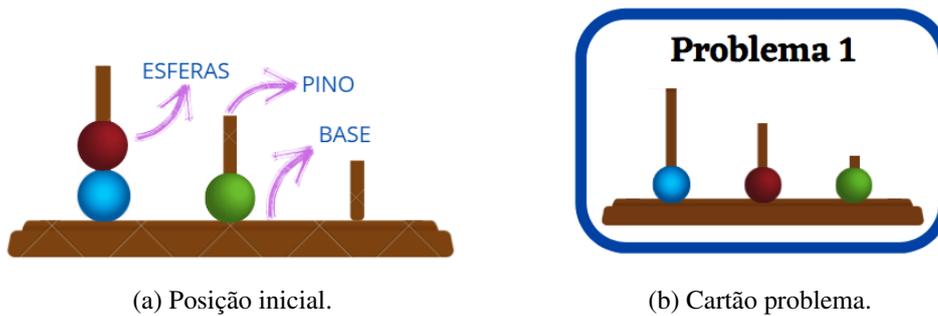


Figura 4.4. Apresentação da ToL: (a) representação tangível do jogo com 3 pinos de tamanhos diferentes, com 3 esferas de mesmo tamanho, demonstrando a posição inicial do jogo; (b) um exemplo de cartão-problema do jogo.

Ao jogador, cabe reproduzir as configurações do cartão problema, Figura 4.4b, o que lhe é dito de antemão, movendo uma esfera de cada vez, com o número mínimo de ações, como exibido na Figura 4.5. Uma resposta correta significa que a solução foi alcançada com o número mínimo de ações [Souza et al. 2001]. A cada rodada um novo cartão-problema é apresentado. Além disso, o jogador sempre sairá da posição inicial e deverá solucionar o problema proposto. Ademais, os cartões problemas têm ordem crescente de dificuldade.

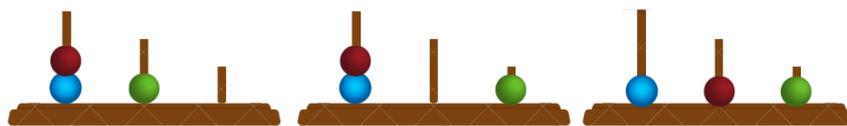


Figura 4.5. ToL: demonstração da solução do jogo partindo da posição inicial e seguindo os passos até chegar no estado final determinado pelo cartão-problema.

A Torre de Londres, dessa forma, é aplicada em tratamentos neurocognitivos, neurológicos e até mesmo neuropsicológicos, sendo pouquíssimas vezes trabalhada e/ou pesquisada para fins computacionais. Portanto, para tal, foi necessário realizar algumas adaptações para que a mesma fosse estudada como um jogo combinatório, com o qual pode se gerar algoritmo e modelagem em grafos.

A mudança realizada foi retirar a questão do jogo vir com uma cartela de problemas. Então, definiu-se que, para iniciar o jogo, deve-se realizar uma distribuição aleatória das esferas, como as apresentadas na Figura 4.6, assim, o estado final do jogo sempre será todas as esferas em p_0 (origem). Sendo que, deve seguir a ordem das cores das esferas (verde, vermelho e azul), como apresentado na Figura 4.7a.

No jogo da ToL, o número mínimo de movimentos depende da distribuição alea-

Estado inicial	M						
	0		3		6		5
	1		7		5		3
	2		7		7		8
	3		6		3		8
	1		4		2		7
	6		6		4		6
	2		4		8		5

Figura 4.6. ToL: possíveis estados iniciais do jogo e o M - número mínimo de movimentos para atingir o estado final do jogo.

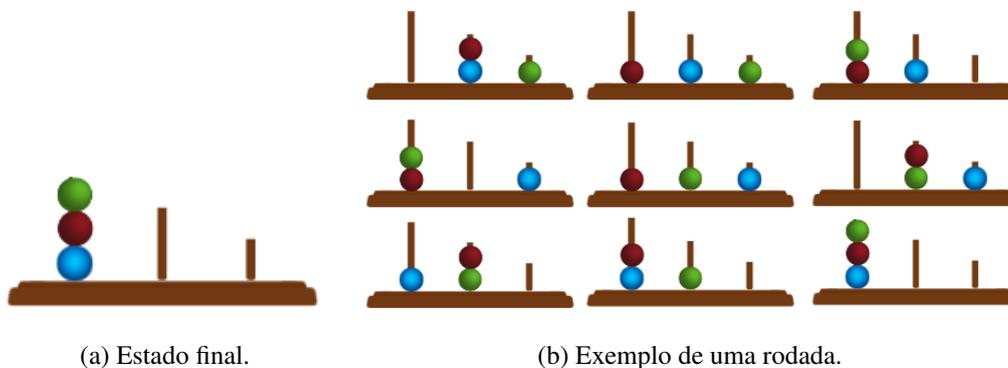


Figura 4.7. ToL: (a) Estado final do jogo definido pela sequência de cores verde, vermelho e azul e (b) demonstração da solução do jogo, mostrando todos os passos, até que o objetivo do jogo seja alcançado.

tória realizada inicialmente. No exemplo abordado na Figura 4.7b, o jogo foi solucionado com 8 movimentos. Na Figura 4.6 são apresentados os números de movimentos mínimos para todas as distribuições possíveis dos discos nos pinos p_0, p_1, p_2 .

4.3.1. Aspectos Algorítmicos

Seja um jogo da ToL com três esferas de cores diferentes, como nos exemplos utilizados neste trabalho: azul, vermelha e verde. De maneira geral, o Algoritmo 7 considera as cores das esferas como índices dos discos, ou seja, d_1 representará a esfera verde, d_2 a esfera vermelha e d_3 a esfera azul. Portanto, o estado final deste será “ d_1, d_2, \dots, d_n ”.

Dado que o jogo da ToL tem como entrada um estado em que os discos se encontram distribuídos de forma aleatória, o Algoritmo 7 inicia verificando onde está o disco d_n , pois, dependendo de sua localização, o jogo será solucionado de forma diferente. Esse al-

goritmo possui internamente 3 funções embutidas para auxiliar na resolução: *desenterra*, *libera* e *move*.

A função **Desenterra** (n, p_i, p_j) é chamada para deixar o disco d_n no topo do seu respectivo pino. Ela recebe 3 parâmetros, o primeiro é índice do disco que se pretende deixar no topo da pilha. O segundo parâmetro é a pilha onde o d_n se encontra. Já o terceiro, é a pilha ao qual os discos que estão “enterrando” d_n devem ser movidos. Então, por exemplo, a função *desenterra* busca o disco d_n na pilha p_0 , movendo todos os discos que estão nesta para p_1 . Quando ela encontra o disco d_n , ela para e retorna a pilha p_0 com o disco d_n no topo.

A função **Libera** (p_i, p_j) é chamada quando se precisa mover algum disco para uma pilha, entretanto, ela precisa estar vazia. Essa função recebe 2 parâmetros. No primeiro, a pilha que precisa estar vazia e o segundo é a pilha que servirá de auxiliar, recebendo os discos que não serão organizados agora. Então, a função *esvazia* p_2 , por exemplo, movendo todos os discos para p_1 .

Por fim, tem-se a função **MoveDisco** (d_n, p_i, p_j) . Essa função é chamada quando se precisa retirar um disco de uma pilha e inserir em outra. Ela recebe 3 parâmetros. No primeiro, o disco d_n , o qual se deseja mover. No segundo parâmetro, a pilha a qual ele se encontra e deve ser removido. No terceiro parâmetro, a pilha que ele deve ser inserido.

Apesar do jogo da ToL ser por padrão apenas com 3 discos (esferas), o Algoritmo [7](#) soluciona o jogo para n discos. Para um número de discos pequenos, tal como 2, 3, ..., 6, ele resolve no menor caminho. Porém, não foi testado para mais de 6 disco, portanto, não pode-se afirmar se é o algoritmo ótimo para solucionar o jogo. Pois, quando a quantidade de discos é grande, não é realizado nenhum tratamento nas pilhas para diminuir o esforço de execução.

Algoritmo 7: ToL - Algoritmo Recursivo

Entrada: d_1, d_2, \dots, d_n e 3 pinos de capacidade diferente ($p_0 = n$ discos, $p_1 = n - 1$ discos e $p_2 = n - 2$ discos). Estado inicial: os n discos distribuídos de forma aleatória.

Saída: Estado final: todos os discos empilhados em p_0 .

```
1 Londres ( $n, p_0, p_1, p_2$ )
2   se ( $d_n \in p_0[0]$ ) então
3     | Londres ( $n - 1, p_0, p_1, p_2$ )
4   senão
5     se ( $d_n \in p_0$ ) então
6       | se ( $d_n \notin p_0[topo]$ ) então
7         | Desenterra( $n, p_0, p_1$ )
8         fim
9       se ( $p_2 \notin \emptyset$ ) então
10        | Libera( $p_2, p_1$ )
11        fim
12        MoveDisco( $d_n, p_0, p_2$ )
13      fim
14      se ( $d_n \in p_1$ ) então
15        | se ( $p_0 \in \emptyset$ ) e ( $d_n \in p_1[topo]$ ) então
16          | MoveDisco( $d_n, p_1, p_0$ )
17          | Londres ( $n - 1, p_0, p_1, p_2$ )
18        senão
19          | se ( $p_2 \notin \emptyset$ ) então
20            | Libera( $p_2, p_0$ )
21            fim
22          Desenterra( $n, p_1, p_0$ )
23          MoveDisco( $d_n, p_1, p_2$ )
24        fim
25      fim
26      se ( $d_n \in p_2$ ) e ( $d_n \notin p_2[topo]$ ) então
27        | Desenterra( $n, p_2, p_1$ )
28        fim
29      Libera( $p_0, p_1$ )
30      MoveDisco( $d_n, p_2, p_0$ )
31      Londres ( $n - 1, p_0, p_1, p_2$ )
32    fim
33 fim
```

4.3.2. Modelagem em Grafos

O jogo da ToL pode ser associado a um grafo $L_p^n = (V, E)$, onde: p é o número de pinos, n o número de esferas, V é o conjunto de vértices com os estados regulares do jogo e E as arestas representando os movimentos permitidos entre os estados regulares.

Obedecendo as regras do jogo da ToL com 2 esferas, tem-se o grafo L_3^2 , conforme a Figura 4.8a. Na ToL com 3 esferas, tem-se o grafo da ToL L_3^3 , conforme a Figura 4.8b. Os estados do jogo são nomeados conforme a Seção 3.3.1 e as propriedades desse grafo são definidas na Proposição 4.

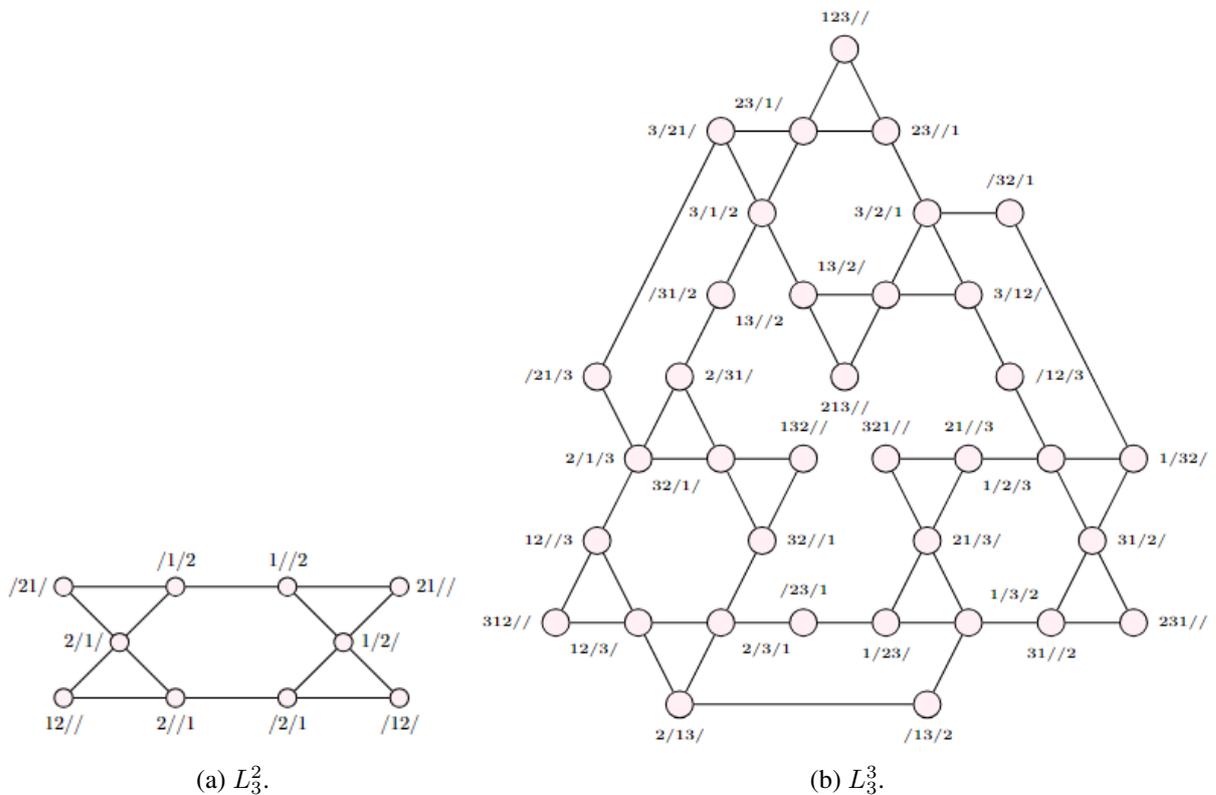


Figura 4.8. Grafos da Torre de Londres.

Proposição 4 (Propriedades do grafo L_3^n). *É um grafo simples, não direcionado, conexo, não é planar ($n \geq 4$) e é composto por subgrafos borboletas.*

O grafo L_3^2 , Figura 4.8a, é um grafo planar e é composto por 2 subgrafos borboleta (*butterfly*), unidos por 2 arestas. L_3^3 , apresentado na Figura 4.8b, é equilibrado em graus no sentido de que tem 12 vértices de cada um dos graus 2, 3 e 4, totalizando 36 vértices, o que pode ser comprovado pelo Teorema 1.

Teorema 1 (Número de vértices de L_3^n). *O grafo L_3^n possui $(\frac{(n+1)(n+2)}{2} - 4)n!$ vértices, quando $n \geq 3$.*

Demonstração. Contar o número de estados do jogo da ToL é possível por meio de sua nomenclatura, contando de quantas maneiras pode ser representado um estado e subtraindo-se os estados irregulares. Então, basta contar de quantas maneiras diferentes é possível inserir as duas “/” entre as esferas do jogo. Considerando inicialmente que todas as esferas são da mesma cor, tem-se $(n + 1)$ formas de inserir a primeira “/”. Então, onde será inserida a segunda “/” depende da posição da primeira. Se a primeira “/” for colocada na posição i , sobra $(n + 1) - i$ posições. Dessa forma, faz-se:

$$\sum_{i=0}^n (n + 1) - i = \sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}$$

Mas, dada a capacidade dos pinos, deve-se retirar os estados que são irregulares no jogo. Retira-se os estados:

$$\begin{aligned} & /d_1, d_2, \dots, d_n/ \\ & //d_1, d_2, \dots, d_n \\ & d_1//d_2, \dots, d_n \\ & /d_1/d_2, \dots, d_n \end{aligned}$$

Portanto é preciso subtrair esses 4 estados. Agora, falta levar em consideração que no jogo as esferas são coloridas, então cada esfera pode estar em qualquer uma dessas posições. Então, multiplica-se todas as possibilidades por $n!$.

$$|V_L| = (\frac{(n+1)(n+2)}{2} - 4)n!$$

Logo, O grafo L_3^n possui $(\frac{(n+1)(n+2)}{2} - 4)n!$ vértices, quando $n \geq 3$. □

Todos os movimentos para a resolução do jogo, no grafo L_3^3 , com o mesmo estado regular inicial demonstrado na Figura 4.7b, são apresentados na Figura 4.9, destacando em amarelo, o caminho percorrido para solucionar o jogo com menor número de movimentos possíveis, respeitando todas as regras.

4.4. Torre de Oxford

O jogo da Torre de Oxford (ToO), como apresentada na Figura 4.10a, possui três discos de mesmo tamanho: $(d_1, d_2$ e $d_3)$ e uma base retangular contendo três pinos de mesmo

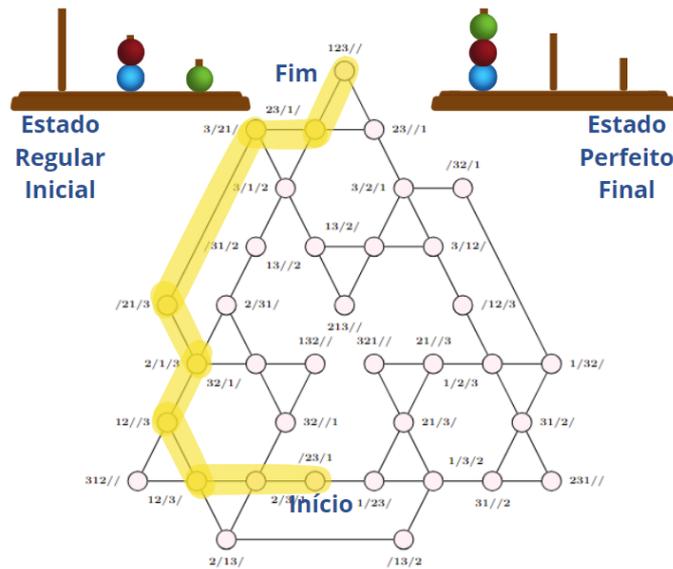


Figura 4.9. Torre de Londres: exemplo do caminho de uma jogada no grafo L_3^3 .

tamanho: $(p_0, p_1$ e $p_2)$. O jogo da ToO foi criado com o objetivo de realizar uma comparação com uma variação semelhante, a ToL no caso. Porém, relaxando algumas restrições na estrutura da ToL e realizar análise dos resultados [Unterrainer et al. 2005] e sua aplicabilidade continuou sendo no estudo de neuropsicologia, tal qual a ToL.

Para resolver o jogo, diferente do jogo clássico da Torre de Hanoi e de muitas variações, na Torre de Oxford, deve-se obedecer apenas uma regra:

- i. mover um disco de cada vez.

O objetivo do jogo é mover os discos de uma posição inicial, como no exemplo apresentado na Figura 4.10a, até chegar à configuração final, como proposto na Figura 4.10b, obedecendo sempre a regra do jogo. Ao jogador, cabe reproduzir a estrutura sugerida como problema, tal qual a anunciada na Figura 4.10b, o que lhe é dito de antemão, movendo um disco de cada vez, com o número mínimo de ações, como exibido na Figura 4.10c. Uma resposta correta significa que a solução foi alcançada com o número mínimo de movimentos.

Assim como na ToL, a cada rodada um novo problema é proposto. O jogador sempre sairá da mesma posição inicial e deverá chegar ao estado do problema proposto. No exemplo abordado na Figura 4.10c, o jogo foi solucionado com 3 movimentos.

Porém, nesta pesquisa, para a Torre de Oxford ser estudada como um jogo combinatório, gerando algoritmo e modelagem em grafos, decidiu-se trabalhar com duas ver-

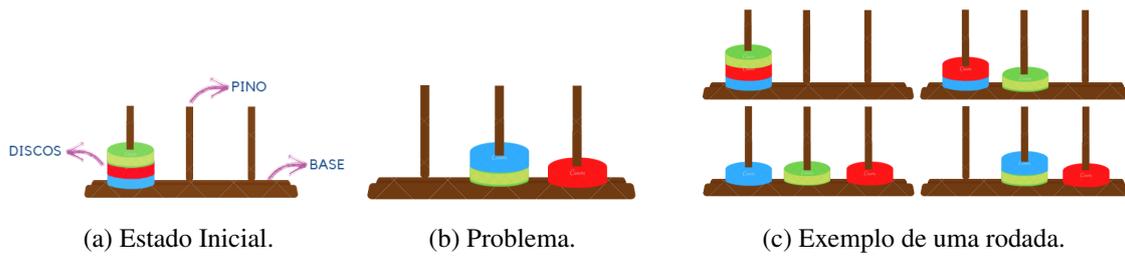


Figura 4.10. Apresentação da Torre de Oxford.

sões dessa torre. Na primeira, definiu-se um estado inicial, como apresentado na Figura 4.11a, e baseando-se na Torre de Hanoi, determinou-se que o objetivo do jogo será levar todos os discos de p_0 (origem), para p_2 (destino), sem se preocupar com a ordem, mas guardando a sua essência que, no caso, é ter apenas uma regra:

- i. mover um disco de cada vez.

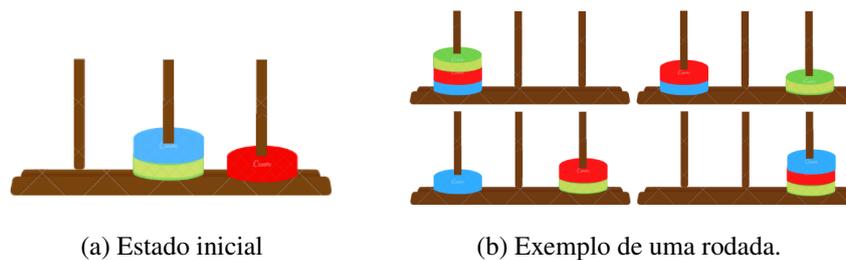


Figura 4.11. Apresentação da Torre de Oxford Versão 1.

Dessa forma, este jogo combinatório torna-se o mais rápido e mais simples de ser resolvido, como mostra a Figura 4.11b. Sendo necessários no máximo n movimentos para solucionar o jogo.

Para a descrição da Torre de Oxford Versão 2, definiu-se que o estado inicial, tal como na ToL, será uma distribuição aleatória dos n discos e um estado final, como na Figura 4.12a e novamente seguindo o modelo da ToL, decretou-se que o objetivo do jogo é levar todos os discos para p_0 , porém na pilha final os discos devem estar na ordem de cores predefinida e permanecendo com a regra de mover apenas um disco por vez.

No jogo da Torre de Oxford versão 2, o número de movimentos depende do n e da distribuição aleatória inicial. Na Figura 4.12b, temos a demonstração do funcionamento do jogo com 3 discos.

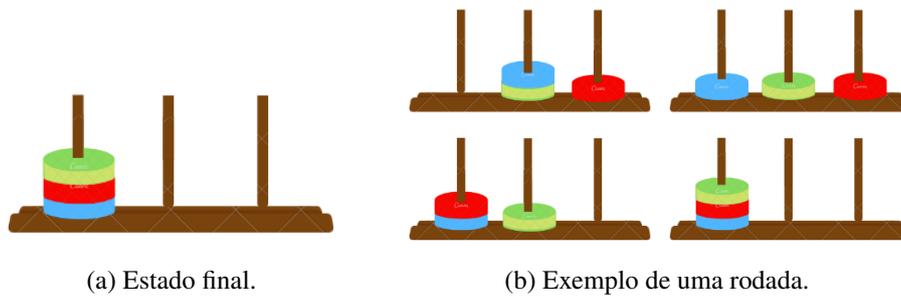


Figura 4.12. Demonstração do jogo da ToO Versão 2.

4.4.1. Aspectos Algorítmicos da Torre de Oxford Versão 1

Para resolver esta versão do jogo, o jogador só precisa mover disco a disco para o pino de destino (p_2), como demonstrado no Algoritmo 8.

Algoritmo 8: Torre de Oxford Versão 1 - Algoritmo Iterativo.	
Entrada:	n discos de tamanhos e cores iguais; e, 3 pinos (p_0, p_1, p_2) de mesma capacidade.
	Estado inicial: todos os n discos empilhados em p_0 .
Saída:	Estado final: todos os n discos empilhados em p_2 .
1	início
2	para $i = 1$ até $i = n$ faça
3	mover d_i de p_0 para p_2
4	fim
5	fim

Ao analisar o algoritmo 8, retira-se a função de custo $T(n) = n$, para $n \geq 1$. Esse algoritmo é trivialmente $O(n)$, portanto, pertence à classe de problemas lineares, apresentando o melhor cenário possível para um jogo combinatório que precisa processar n discos.

4.4.2. Aspectos Algorítmicos da Torre de Oxford Versão 2

O algoritmo de resolução da Torre de Oxford Versão 2, como exibido no Algoritmo 9 é muito similar ao Algoritmo 7 da ToL, porém com menos restrições, dado que na ToO os pinos têm a mesma capacidade.

Desse modo, a principal diferença entre os Algoritmos 7 e 9 é quando d_n está em p_1 . No Algoritmo 7, d_n tem que ser movido para p_2 para p_1 funcionar como auxiliar para liberar p_0 , pois em p_2 não é garantido que caberá os d_1, \dots, d_{n-1} discos. Já no Algoritmo

9, não há esse problema, pois todos os pinos suportam os d_1, \dots, d_n discos. Então, os d_1, \dots, d_{n-1} discos são movidos para p_2 e d_n é inserido em p_0 diretamente.

Algoritmo 9: ToO - Algoritmo Recursivo

Entrada: d_1, d_2, \dots, d_n e 3 pinos de mesma capacidade (p_0, p_1 e p_2). Estado inicial: os n discos distribuídos de forma aleatória.

Saída: Estado final: todos os discos empilhados em p_0 .

```

1 Oxford ( $n, p_0, p_1, p_2$ )
2   se ( $d_n \in p_0[0]$ ) então
3     |   Oxford ( $n - 1, p_0, p_1, p_2$ )
4   senão
5     |   se ( $d_n \in p_0$ ) então
6       |   se ( $d_n \notin p_0[topo]$ ) então
7         |   Desenterra( $n, p_0, p_1$ )
8         fim
9         MoveDisco( $d_n, p_0, p_2$ )
10      fim
11     |   se ( $d_n \in p_1$ ) então
12       |   se ( $p_0 \in \emptyset$ ) e ( $d_n \in p_1[topo]$ ) então
13         |   MoveDisco( $d_n, p_1, p_0$ )
14         |   Oxford ( $n - 1, p_0, p_1, p_2$ )
15       |   senão
16         |   se ( $p_0 \notin \emptyset$ ) então
17           |   Libera( $p_0, p_2$ )
18           fim
19         |   Desenterra( $n, p_1, p_2$ )
20         |   MoveDisco( $d_n, p_1, p_0$ )
21         |   Oxford ( $n - 1, p_0, p_1, p_2$ )
22       |   fim
23     |   fim
24     |   se ( $d_n \in p_2$ ) e ( $d_n \notin p_2[topo]$ ) então
25       |   Desenterra( $n, p_2, p_1$ )
26     |   fim
27     |   Libera( $p_0, p_1$ )
28     |   MoveDisco( $d_n, p_2, p_0$ )
29     |   Oxford ( $n - 1, p_0, p_1, p_2$ )
30   fim
31 fim

```

4.4.3. Modelagem em Grafos

Para modelagem dos grafos, a partir daqui, quando se falar em Torre de Oxford (ToO) estará sendo considerada a versão 2, explorada anteriormente. Versão essa escolhida devido ao seu maior grau de dificuldade.

O jogo da ToO tem número finito de posições e, portanto, pode ser associado a um grafo $O_p^n = (V, E)$, em que p é o número de pinos do jogo, n é o número de discos, V são os vértices que representam os estados regulares do jogo e E as arestas que são os movimentos permitidos entre os estados. Os vértices são nomeados conforme a Seção 3.3.1 e as propriedades desse grafo são definidas na Proposição 5.

Proposição 5 (Propriedades do grafo O_3^n). *É um grafo simples, não direcionado, não planar ($n \geq 3$), conexo e é composto por subgrafos borboletas.*

Para fundamentar a Proposição 5, começa-se analisando o jogo da ToO com apenas 1 disco. Esse grafo tem 3 estados possíveis, igual ao jogo da ToH, como mostrado na Figura 3.3b. Pode-se movimentar o disco para qualquer pino. Portanto, a representação de O_3^1 fica como na Figura 3.3b, um grafo C_3 .

Na ToO com 2 discos, tem-se o grafo O_3^2 , como apresentado na Figura 4.13a, o 1 corresponde ao disco da cor azul e o 2 refere-se ao disco vermelho. O grafo O_2 é planar e fortemente conexo, tem ordem 12 e é equilibrado em graus, pois possui 6 vértices de grau 2 e 6 de grau 4. Acrescentando mais um disco ao jogo da Torre de Oxford, pode-se observar, na Figura 4.13b, que o grafo O_3^3 possui 18 vértices de grau 2, 36 vértices de grau 4 e 6 vértices de grau 6 e sua ordem aumenta bruscamente, passando de 12 para 60, o que é demonstrado no Teorema 2.

Teorema 2 (Número de vértices do O_3^n). *O grafo O_3^n possui $\left(\frac{(n+1)(n+2)}{2}\right)n!$ vértices, quando $n \geq 3$.*

Demonstração. Assim como na ToL, pode-se encontrar o número de vértices do grafo O_3^n , basta contar de quantas formas pode ser representado um estado. Assim, precisa-se saber de quantas maneiras diferentes é possível inserir as duas “/” entre os discos do jogo. Considerando, inicialmente, que todos os discos são da mesma cor, têm-se $(n + 1)$ formas de inserir a primeira “/”. Então, onde será inserida a segunda “/” dependerá da posição

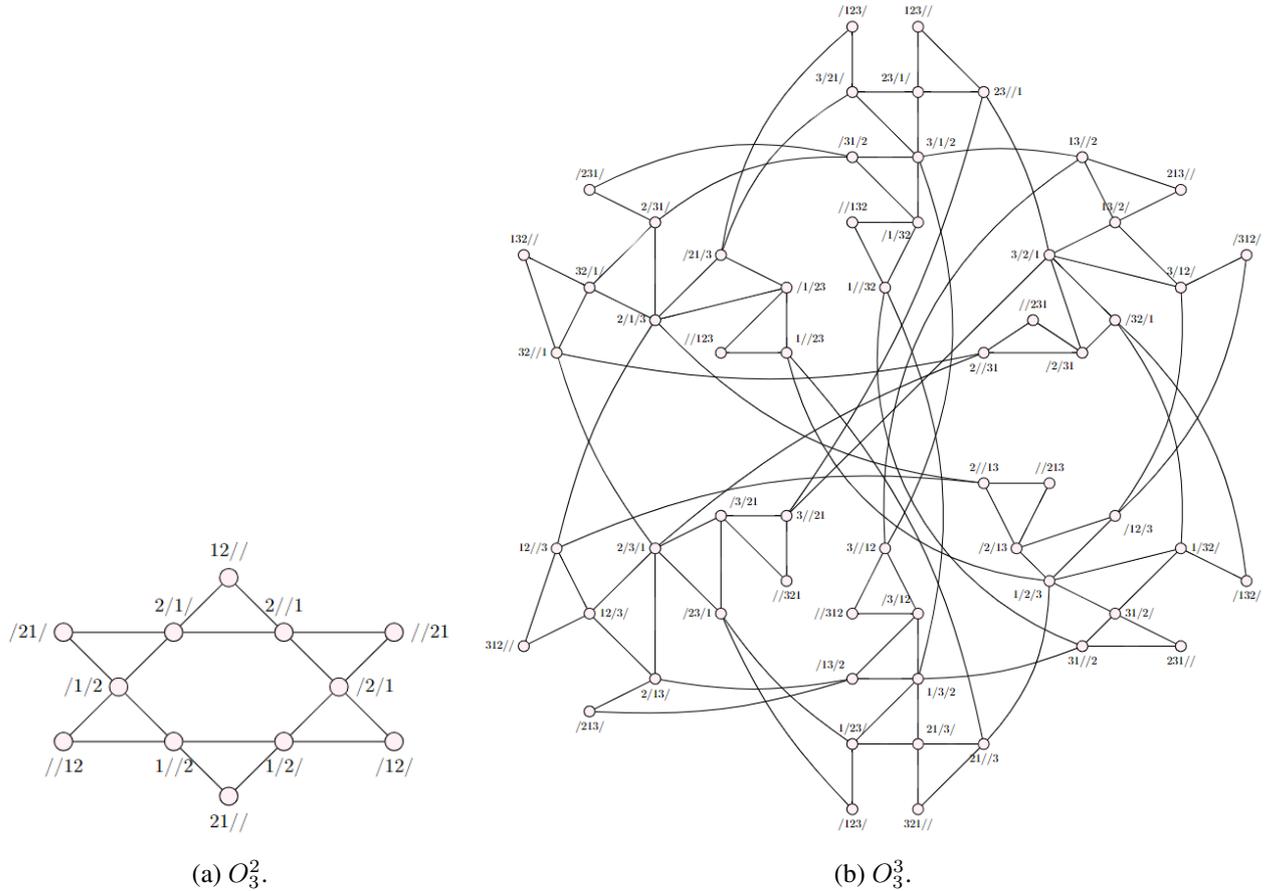


Figura 4.13. Grafos da Torre de Oxford.

da primeira. Se a primeira “/” for colocada na posição i , sobra $(n + 1) - i$ posições. Dessa forma, faz-se:

$$\sum_{i=0}^n (n + 1) - i = \sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}$$

Como a ToO não possui restrição nos pinos, agora só falta levar em consideração que no jogo os discos são de cores diferentes. Então cada disco pode estar em qualquer uma dessas posições. Assim, multiplica-se todas as possibilidades por $n!$.

$$|V_O| = \left(\frac{(n+1)(n+2)}{2}\right)n!$$

Logo, O grafo O_3^n possui $\left(\frac{(n+1)(n+2)}{2}\right)n!$ vértices, quando $n \geq 3$. □

O desenho do grafo O_3^n apresentado na Figura 4.13b não é planar, mas isso não é uma característica apenas desse desenho como se pode ver pelo Teorema 3.

Teorema 3 (Sobre a planaridade do O_3^n). *O grafo O_3^n não é planar [Hinz et al. 2009], para todo $n \geq 3$.*

Demonstração. O Teorema de Kuratowski [Diestel 2017] mostra que um grafo é planar, se e somente se, ele não possui um subgrafo que seja uma subdivisão de k_5 ou $k_{(3,3)}$. Dado que o grafo O_3^3 , Figura 4.13b, pode ser representado como na Figura 4.14. Assim, considerando cada componente induzido pelos vértices de graus 2 e 4 do grafo (destacadas em amarelo na Figura 4.14) como um super-vértice, é possível observar que cada vértice de grau 6 do grafo O_3^3 é conectado aos 3 super-vértices. Essa estrutura é mais simples e já não é planar, pois possui um $K_{3,3}$ como subgrafo. Como em geral o O_3^n tem O_3^{n-1} como subgrafo, o O_3^n possui essa estrutura não planar para todo $n \geq 3$.

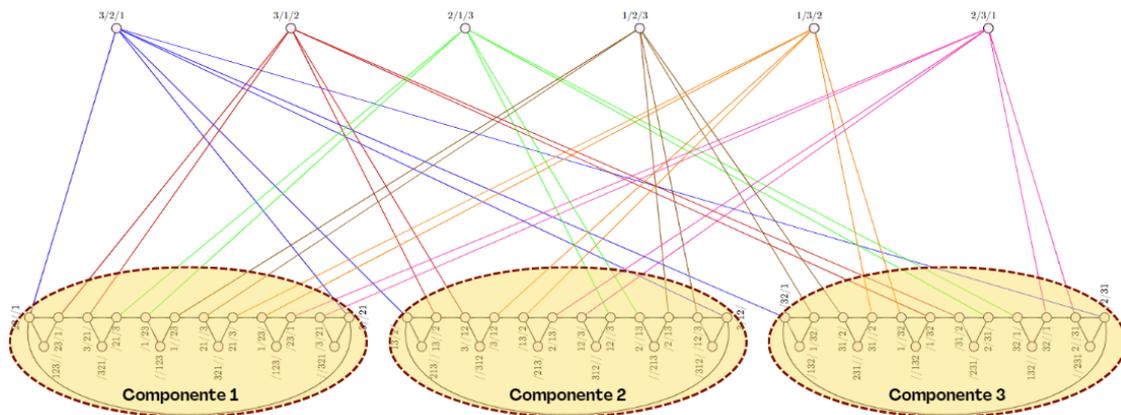


Figura 4.14. Grafo O_3^3 .

□

No jogo da ToL com n discos, se 2 discos forem fixados na base de p_0 , os demais $n - 2$ poderão ser movidos livremente sem violar nenhuma capacidade. Isso implica que O_3^n é subgrafo de L_3^{n+2} . Logo, do Teorema 3 segue o Corolário 1.

Corolário 1 (Sobre a planaridade do L_3^n). *O grafo L_3^n não é planar [Hinz et al. 2009], para todo $n \geq 4$.*

Um exemplo do jogo da ToO com 3 discos é dado na Figura 4.15, onde pode-se observar o estado inicial e o estado perfeito final do jogo. Em amarelo está evidenciado o caminho percorrido pelo jogador, demonstrando os 5 movimentos realizados, ou seja, o número mínimo de movimentos necessários para solucionar o jogo.

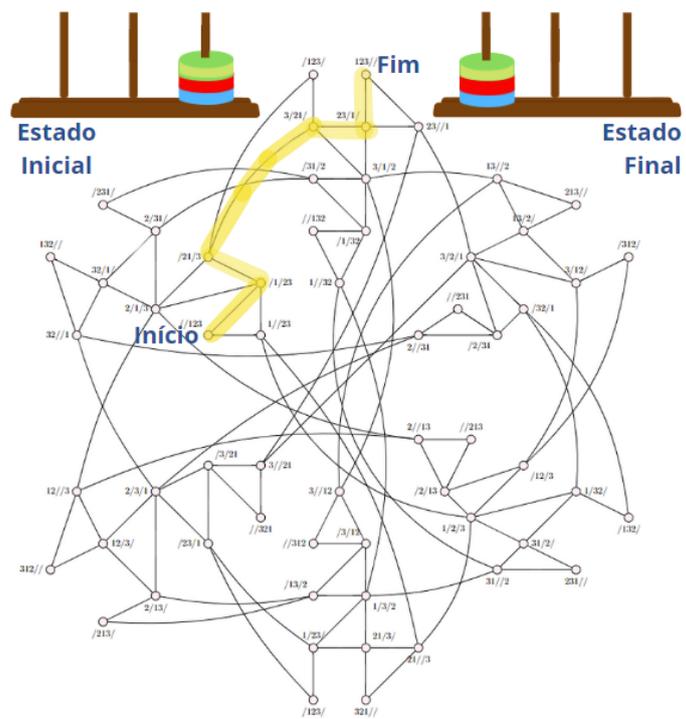


Figura 4.15. ToO: exemplo do caminho percorrido para solucionar o jogo no grafo O_3^3 .

Capítulo 5

Proposta do Jogo Torre de Manaus

Neste capítulo é apresentada a nova variação da ToH, a Torre de Manaus. Inicia-se explorando sua estrutura, jogabilidade e regras. Depois, têm-se os algoritmos para a solução do jogo. Seguido das propriedades do grafo desse jogo e a sua caracterização. Continuando, expõe-se as aplicações do jogo e, por fim, enfatiza-se os trabalhos relacionados com essa nova variação.

5.1. Unindo Características dos Jogos da Torre de Hanoi e da Torre de Londres

O jogo da Torre de Manaus (ToM) consiste em n discos d_1, d_2, \dots, d_n com diferentes tamanhos empilhados aleatoriamente em 3 pinos p_0, p_1, p_2 . Os discos em cada pino, de baixo para cima, são dispostos em ordem decrescente de tamanho, e p_0, p_1, p_2 têm capacidade máxima $n, n - 1, n - 2$ respectivamente. O objetivo é mover os n discos para p_0 [Martins e Freitas 2023], como mostra a Figura 5.1a.

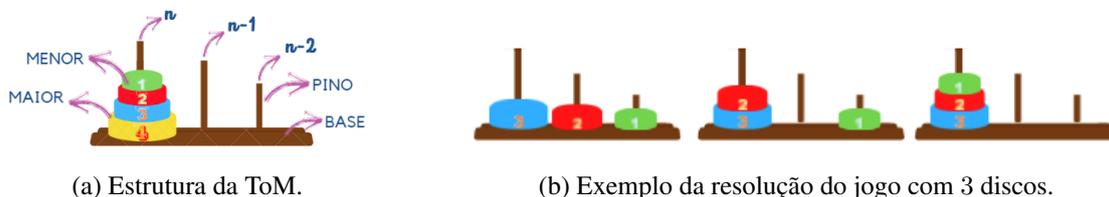


Figura 5.1. Torre de Manaus: (a) representação tangível do jogo com 4 discos, com o menor disco, na cor verde, no topo da pilha e o maior disco, na cor amarela, na base; (b) demonstração da solução do jogo com 3 discos, partindo de um estado regular inicial e empilhando todos os discos em ordem decrescente de tamanho em p_0 .

Quanto às regras do jogo da ToM, têm-se:

- i. em cada movimento, somente o topo de um dos pinos pode ser movido;
- ii. o disco d_i só pode ser colocado sobre d_j , se $i < j$;
- iii. um disco só pode ser movido para um pino, se esse pino, não tiver atingido a sua capacidade máxima. A capacidade de $p_0 = n$, $p_1 = n - 1$ e $p_2 = n - 2$ discos.

No jogo da ToM, um estado regular é um estado em que nenhuma regra do jogo está sendo violada, o estado inicial é qualquer estado regular, o estado perfeito final são todos os discos empilhados em p_0 , tendo os discos em ordem decrescente de tamanhos e um estado irregular é um estado em que pelo menos uma regra é desrespeitada. Nesse jogo, o jogador parte de um estado regular qualquer e deve terminar sempre no estado perfeito final. A Figura 5.2 trás todos os estados regulares possíveis para o jogo ToM com 3 discos.

Estado Inicial	M						
	0				3		3
	1		2				3
	2						
	4						
	1		5				

Figura 5.2. ToM com 3 discos apresentando todas as possibilidades de estados regulares iniciais do jogo. Sendo que, a coluna M mostra o número mínimo de movimentos necessários para o jogador que iniciar o jogo nesse estado possa solucionar o jogo ou o símbolo vermelho que indica que partindo daquele estado, não é possível solucionar o jogo sem violar as regras.

Dadas as regras, o número mínimos de discos para praticar o jogo são 2. Pode-se visualizar na Figura 5.1b, um exemplo do jogo da ToM, partindo de um estado inicial e finalizando com o empilhamento de todos os discos no primeiro pino. Apesar do jogo ter 3 discos, dado o estado inicial, ele precisa de apenas 2 movimentos para ser concluído.

O número mínimo de movimentos para solucionar o jogo depende do número de discos e do estado regular que inicia o jogo, conforme a Figura 5.2, que apresenta o número de movimentos para todos os possíveis estados regulares da ToM com 3 discos. Sendo que, existem alguns estados regulares que, respeitando as três regras do jogo, não

é possível alcançar o estado perfeito final, portanto, os estados regulares estão divididos em bons ou ruins, conforme a Definição 4 e a Definição 5.

Definição 4 (Estado bom). *Um estado bom é um estado regular do jogo em que pode-se chegar ao estado perfeito final.*

Definição 5 (Estado ruim). *Um estado ruim é um estado regular, mas, que partindo-se dele não é possível chegar ao estado perfeito final do jogo sem quebrar as regras.*

Os estados iniciais possíveis para o jogo da ToM com 4 discos são todos apresentados na Figura 5.3. Explorando os estados ruins observa-se nas Figuras 5.2 e 5.3 que todas as vezes que o disco maior está em p_1 o jogo não tem solução. Isso é verdade de maneira geral, como demonstrado no Teorema 2.

Estado Inicial	M	Estado Inicial	M								
	0		3		7		4		9		10
	1		3		7		6		10		11
	1		3		7		7		11		11
	2		4		9		7		11		11
	3		5		5		7		10		10
	2		6		5		8		8		8
	0		3		7		4		9		10

Figura 5.3. ToM com 4 discos apresentando todas as possibilidades de estados regulares iniciais do jogo. Sendo que, a coluna M mostra o número mínimo de movimentos necessários para o jogador que iniciar o jogo nesse estado possa solucionar o jogo ou o símbolo vermelho que indica que partindo daquele estado, não é possível solucionar o jogo sem violar as regras.

Lema 2 (Condição para ser um estado ruim). *Na ToM, se no estado inicial do jogo $d_n \in p_1$, então, esse estado é ruim, para todo $n \geq 2$.*

Demonstração. Para resolver uma ToM com n discos, se d_n está em p_1 é necessário movê-lo para p_0 . Supondo que isso seja possível, considere o movimento que leva d_n para p_0 . Para tal, têm-se duas formas:

1º caso: se d_n está em p_1 e p_2 está vazio. Para mover d_n para p_0 os discos d_1, d_2, \dots, d_{n-1} devem ser inseridos todos em p_2 , mas isso não é possível, uma vez que viola a capacidade de p_2 ;

2º caso: se d_n está em p_1 e p_0 está vazio. Para mover d_n para p_0 diretamente sem passar por p_2 , seria necessário que todos os discos d_1, d_2, \dots, d_{n-1} fossem movidos para p_2 , como no 1º caso. Outra forma, seria passando d_n por p_2 . Então, para mover d_n para p_2 é necessário mover os discos d_1, d_2, \dots, d_{n-1} para p_0 , mover d_n para p_2 e depois os discos d_1, d_2, \dots, d_{n-1} para p_1 . Porém, nesse caso, para mover o disco d_{n-1} de p_0 para p_1 , é necessário que todos os outros discos estejam em p_2 , violando assim sua capacidade.

Logo, não é possível mover d_n para p_0 . □

Lema 3 (Condição para ser um estado bom). *Na ToM, se no estado inicial do jogo $d_n \in p_0$, então, esse estado é bom, para todo $n \geq 2$.*

Demonstração. Seja um estado da ToM, onde $d_n \in p_0$. Como d_n já está em p_0 só os discos d_1, d_2, \dots, d_{n-1} precisam ser movidos para p_0 , para solucionar o jogo. Então, analise-se 3 casos: $d_{n-1} \in p_0$, $d_{n-1} \in p_1$ ou $d_{n-1} \in p_2$:

No 1º caso, como d_{n-1} já está em p_0 , para mover todos os discos d_1, d_2, \dots, d_{n-2} para p_0 não haverá restrição nenhuma, pois todos os pinos terão capacidade para receber esse conjunto de discos, se for necessário no decorrer do processo, até chegar a solução;

No 2º caso, o disco d_{n-1} está em p_1 e precisa ser movido para p_0 . Antes de mover d_{n-1} a primeira vez, o mesmo precisa ser liberado. Para isso, somente $(n - 2)$ discos precisam ser manipulados até que d_1, d_2, \dots, d_{n-2} estejam em p_2 , o que não viola a capacidade de p_2 . Após mover d_{n-1} para p_0 volta-se ao 1º caso;

No 3º caso, o d_{n-1} está em p_2 . Suponha que para chegar a solução do jogo seja necessário acrescentar k discos em p_2 . Considere que k é o maior número de discos que será acrescentado em p_2 . Se é necessário mover k discos para p_2 , obrigatoriamente não é possível usar p_0 ou p_1 como auxiliar. Isso implica que tem um disco fora de p_2 diferente de d_n impedindo de utilizar p_0 ou p_1 como auxiliar. Pois, como d_{n-1} está em p_2 , tanto p_0 quanto p_1 teriam capacidade suficiente para acrescentar k discos. Logo, $k \leq n - 3$ e a capacidade de p_2 não será excedida com a manipulação destes discos.

Portanto, se $d_n \in p_0$ o jogo sempre terá solução. □

Teorema 4 (Caracterização dos estados regulares). *Seja s um estado regular da ToM com n discos. Se $d_n \in p_0$, s é bom. Se $d_n \in p_1$, s é ruim. Se $d_n \in p_2$, seja d_i , o maior disco fora de p_2 , então:*

- i) s é bom se $d_i \in p_0$ e $n - i$ é par ou se $d_i \in p_1$ e $n - i$ é ímpar;*
- ii) s é ruim se $d_i \in p_0$ e $n - i$ é ímpar ou se $d_i \in p_1$ e $n - i$ é par.*

Demonstração. Os casos em que d_n está em p_0 ou p_1 seguem dos Lemas 2 e 3. Se $d_n \in p_2$ para demonstrar que as assertivas *i*) e *ii*) são válidas, têm-se três casos a seguir.

No 1º caso, considere que $i > 2$ e que existe pelo menos um espaço disponível em p_2 . Suponha que d_n está em p_2 e $j \in \{0, 1\}$, seja o índice do pino em que d_i se encontra. Neste caso, é possível colocar os discos d_1, d_2, \dots, d_{i-1} sobre d_i em p_j de modo que d_{i+1} esteja no topo de p_2 , e, p_k esteja vazio, onde $k = 1 - j$. Dessa forma, é possível mover d_{i+1} para p_k e, agora, o maior disco fora de p_2 é d_{i+1} e há pelo menos um espaço disponível em p_2 , ou seja, também é possível colocar os discos d_1, d_2, \dots, d_i sobre d_{i+1} em p_k . Repetindo todo o processo, todos os discos serão movidos para o mesmo pino e se, ao final, todos estiverem em p_0 , o estado inicial é bom. Caso contrário, se todos estiverem em p_1 , o estado inicial é ruim. Observe que quando o estado inicial é bom os discos estarão em p_0 se $j = 0$ e $n - i$ é par ou estarão em p_1 se $j = 1$ e $n - i$ é ímpar, conforme a assertiva *i*). Nos casos opostos, isto é, quando o estado é ruim, $j = 0$, $n - i$ é ímpar ou $j = 1$ e $n - i$ é par, conforme assertiva *ii*).

No 2º caso, $i > 2$ e p_2 está inicialmente cheio. Existem duas possibilidades: na primeira, os dois discos fora de p_2 estão em p_j , $j \in \{0, 1\}$, isso quer dizer que é possível mover o topo de p_2 para p_k , $k = 1 - j$, e, na segunda possibilidade, d_i está sozinho em um pino, e pode-se mover o disco menor que d_i , que está fora de p_2 , para cima de d_i para alcançar o mesmo estado da primeira possibilidade. Isso atinge o 1º caso desta prova, onde há um espaço vazio disponível em p_2 . Deste modo, se conclui *i*) e *ii*) também.

No 3º caso, $i = 2$ e p_2 está inicialmente cheio. Assim como no 2º caso, existem duas possibilidades: na primeira, d_1 e d_2 estão em p_j , $j \in \{0, 1\}$, isso quer dizer que é possível mover d_3 para p_k , $k = 1 - j$, e, na segunda possibilidade, d_2 está sozinho, e pode-se mover d_1 para cima dele para alcançar o mesmo estado da primeira possibilidade. Isso também atinge o 1º caso desta prova, sendo que $d_i = d_3$, portanto, $i > 2$. Deste modo, também se conclui *i*) e *ii*). □

5.2. Aspectos Algorítmicos

Obedecendo as regras, para jogar com a ToM é importante saber se o estado inicial é um estado bom do jogo, ou seja, saber se o jogo tem ou não solução. Para tal, elaborou-se a função exposta no Algoritmo [10](#), que retorna “SIM” caso o estado seja bom ou “NÃO” caso o estado do jogo seja ruim.

Algoritmo 10: Torre de Manaus - Identificação do Estado.	
	Entrada: d_1, d_2, \dots, d_n ; e 3 pinos de capacidade diferente ($p_0 = n$ discos, $p_1 = n - 1$ discos e $p_2 = n - 2$ discos). Estado inicial: estado regular (discos distribuídos de forma aleatória).
	Saída: "SIM"o estado é bom ou "NAO"o estado é ruim.
1	ehBom (n, p_0, p_1, p_2)
2	se ($d_n \in p_0$) então
3	retorna “SIM”
4	senão
5	se ($d_n \in p_1$) então
6	retorna “NAO”
7	senão
8	retorna ehBom ($n - 1, p_1, p_0, p_2$)
9	fim
10	fim
11	fim

Observe que o Algoritmo [10](#) tem como direcionamento a posição de d_n no jogo. Dado o Teorema [4](#), se d_n está em p_1 esse estado é ruim. Então, o Algoritmo [10](#) verifica se d_n está em p_0 ou p_1 , nesse caso pára e retorna. Se d_n está em p_2 , ele chama novamente a função **ehBom**($n - 1, p_1, p_0, p_2$), passando como o maior disco agora o d_{n-1} , pois, quando d_n está em p_2 , no estado inicial, quem determina se o estado é bom é o pino onde d_{n-1} se encontra. Nesse caso, para ser um estado bom, d_{n-1} precisa estar em p_1 , por isso inverte-se a posição de p_0 e p_1 na chamada da função.

Dado que a entrada do algoritmo é um estado regular do jogo, nesse cenário, em p_2 só terá no máximo $n - 2$ discos. Então, sempre existirá pelo menos 2 discos fora de p_2 . No pior caso, a posição do disco d_2 será o critério de parada e definirá se o estado é bom ou não e encerrará o algoritmo.

Após descobrir se o estado inicial é bom, a função para solucionar o jogo está

implementada no Algoritmo [11](#). Essa função recebe o estado bom, que já foi verificado no Algoritmo [10](#), e inicia verificando se $n = 1$, em caso verdadeiro, move d_1 para p_0 . Se n é diferente de 1, verifica se d_n está em p_1 . Em caso afirmativo, chama a função novamente, mas agora passando $(n - 1)$ discos e passa-se p_2 como destino e p_0 como auxiliar. Seguindo, verifica se d_n está em p_2 . Em caso positivo, chama a função **solucaoManaus** $(n - 1, p_1, p_0, p_2)$, passando $(n - 1)$ discos, mas com p_1 como destino e p_0 como auxiliar. Por fim, move-se o disco d_n para p_0 .

Algoritmo 11: Torre de Manaus - Resolução de Um Estado Bom.

Entrada: d_1, d_2, \dots, d_n ; e 3 pinos de capacidade diferente: destino - p_0 : n discos, auxiliar - $p_1 = (n - 1)$ discos e auxiliar - $p_2 = (n - 2)$ discos). Estado inicial: estado bom.

Saída: Estado perfeito final: todos os discos empilhados em p_0 (destino).

```

1 solucaoManaus (n, p0, p1, p2)
2   se (n == 1) então
3     |   move(d1, p0)
4     |   retorna
5   fim
6   se (dn ∈ p1) ou (dn ∈ p0) então
7     |   solucaoManaus (n - 1, p2, p1, p0)
8   fim
9   se (dn ∈ p2) então
10    |   solucaoManaus (n - 1, p1, p0, p2)
11  fim
12  move(dn, p0)
13  solucaoManaus (n - 1, p0, p1, p2)
14 fim

```

$$T(n) = \begin{cases} O(1), & \text{se } n=1 \\ 2T(n-1) + O(1), & \text{se } n>1 \end{cases}$$

O Algoritmo [11](#) tem a equação de recorrência acima. Então, conclui-se que a complexidade de tempo desta função é a mesma da ToH, ou seja, $O(2^n)$. Portanto, o algoritmo deste jogo também é exponencial.

O Algoritmo [12](#), por meio das funções apresentadas no Algoritmo [10](#) e no Algoritmo [11](#), soluciona o jogo. Inicialmente recebe um estado regular, verifica se o estado é bom, se “SIM”, soluciona o jogo. Senão, informa que é “IMPOSSÍVEL”.

Algoritmo 12: Torre de Manaus - Algoritmo Recursivo.

Entrada: d_1, d_2, \dots, d_n ; e 3 pinos de capacidade diferente ($p_0 = n$ discos, $p_1 = n - 1$ discos e $p_2 = n - 2$ discos). Estado inicial: estado regular (discos distribuídos de forma aleatória).

Saída: Estado perfeito final, caso o estado inicial seja bom, ou mensagem "IMPOSSIVEL" caso contrário.

```

1 Manaus ( $n, p_0, p_1, p_2$ )
2   se ( $ehBom(n, p_0, p_1, p_2) == "SIM"$ ) então
3     solucaoManaus( $n, p_0, p_1, p_2$ )
4   senão
5     imprime "IMPOSSIVEL"
6   fim
7 fim

```

5.3. Modelagem em Grafos

O jogo da Torre de Manaus pode ser associado a um grafo $M_p^n = (V, E)$, em que p é o número de pinos, n é o número de discos, V são os estados do jogo e E são as transições possíveis entre os estados. Adicionalmente, os vértices de V são particionados em: V_b e V_r , onde V_b são os vértices bons, ou seja desse estado é possível alcançar o estado final do jogo¹ e V_r são os vértices ruins, partindo deles não é possível chegar ao estado perfeito final do jogo².

Para representar o grafo da ToM, os rótulos dos vértices foram definidos conforme descrito na Seção 3.3.1. Esses rótulos consistem em uma sequência de números em que: o primeiro número indica o disco que está no topo da pilha; a barra "/" separa os números que representam os discos no próximo pino/pilha; e o padrão continua dessa maneira para os demais pinos.

O grafo M_3^1 não existe, pois como dito anteriormente, o jogo ToM só é possível de ser jogado com no mínimo 2 discos, devido às suas regras. Então, para embasar a proposição 6, parte-se da análise do grafo M_3^2 apresentado na Figura 5.4a.

Proposição 6 (Propriedades do grafo M_3^n). *É simples, planar e desconexo.*

Ressaltando o que foi indicado na Proposição 6, a Figura 5.4a apresenta o grafo

¹Nas ilustrações, os V_b aparecem na cor verde.

²Nas ilustrações, os V_r são representados na cor vermelha.

M_3^2 , onde observa-se que ele é simples, tem ordem 3, o grau máximo 1 e o mínimo é 0. Adicionalmente, investigando o grafo do jogo com 3 discos, M_3^3 da Figura 5.4b, possui ordem 19, sendo que, possui 10 vértices com estados bons e 9 vértices com estados ruins. M_3^3 tem o grau máximo 3 e o mínimo 1. Analisando o grafo do jogo com 4 discos - M_3^4 , apresentado na Figura 5.5, possui as mesmas características do grafo M_3^3 , exposto na Figura 5.4b, e a principal delas é que este é um grafo desconexo, ou seja, ele sempre terá 2 componentes, o que é demonstrado no Teorema 5. O grafo M_3^n é um subgrafo de H_3^n e herda as suas características. Portanto, M_3^n é simples, planar e desconexo.

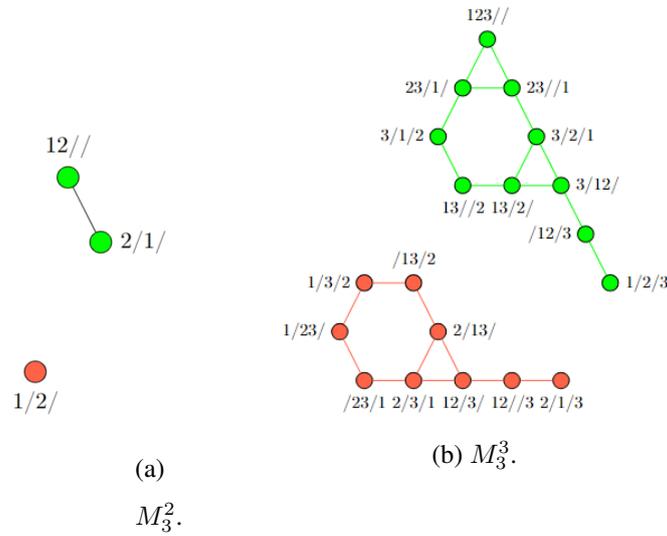


Figura 5.4. ToM: em (a) M_3^2 e (b) M_3^3 , de verde os vértices que representam estados bons e de vermelho os vértices que representam estados ruins.

Teorema 5 (Quantidade de componentes conexas em M_3^n). *O grafo da Torre de Manaus com n discos sempre terá 2 componentes, quando $n \geq 2$.*

Demonstração. Por definição, todos os estados bons estão em uma componente. Precisa-se verificar se todos os estados ruins estão na mesma componente. Primeiramente, mostra-se que existe uma sequência de movimentos para ir de qualquer estado ruim em que d_n está em p_2 para o estado $d_1, d_2, \dots, d_{n-1} // d_n$, como por exemplo, o estado apresentado na Figura 5.6f.

Seja um estado ruim em que d_i é o maior disco fora de p_2 . Primeiramente, considere o caso em que $d_i > d_2$ (Figura 5.6a). Se d_i está em p_0 e há espaço livre em p_2 , é possível mover os discos d_1, d_2, \dots, d_i para p_0 (Figura 5.6b) e depois mover d_{i+1} para p_1 (Figura 5.6c).

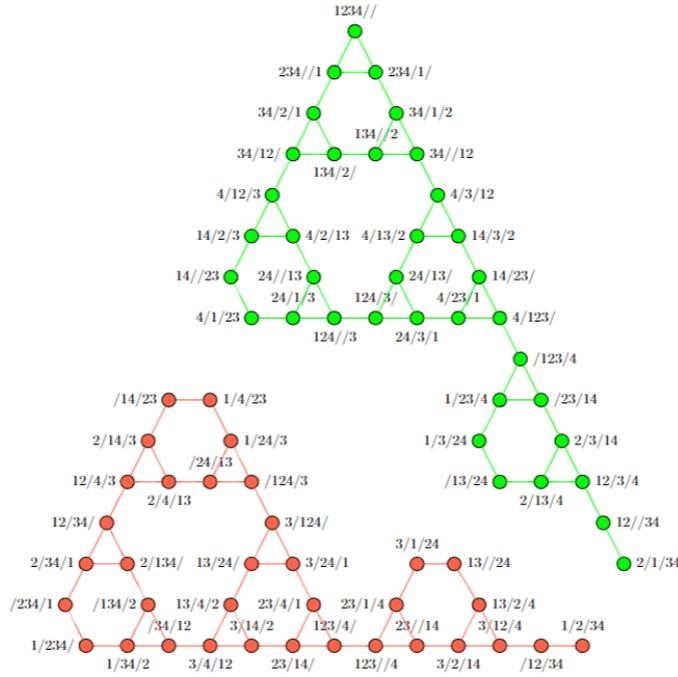


Figura 5.5. M_3^4 - Grafo da ToM com 4 discos, de verde os vértices que representam os estados bons e de vermelho os vértices que representam estados ruins.

Agora, o maior disco fora de p_2 é d_{i+1} e há espaço em p_2 (Figura 5.6c), então é possível repetir o processo e mover todos os i discos menores para p_1 (Figura 5.6d). Isso é repetido até que os discos d_1, d_2, \dots, d_{n-1} estejam todos empilhados no mesmo lugar (Figura 5.6f). Se eles estão em p_0 ou p_1 depende apenas da paridade de $n - i$, porque o processo alterna entre p_0 e p_1 . Se todos os discos estão em p_1 , então seria possível mover d_n para p_0 e este seria um estado bom. Assim, como o estado inicial é ruim, conclui-se que $n - i$ é ímpar e o processo terminou no estado $d_1, d_2, \dots, d_{n-1} // d_n$ (Figura 5.6f). Por simetria, quando d_i está inicialmente em p_1 (Figura 5.7a), segue-se que $n - i$ deve ser par e o estado $d_1, d_2, \dots, d_{n-1} // d_n$ também será alcançado.

Resta considerar o caso em que o maior disco fora de p_2 é d_2 . Se d_1 e d_2 estiverem no mesmo pino, é possível mover d_3 para o pino vazio e se d_2 estiver sozinho, é possível mover d_1 para cima dele para alcançar aquele estado, e assim por diante. Isso atinge o caso anterior em que o maior disco fora de p_2 é d_3 . Desse modo, se n for par, $d_1/d_2/d_3, \dots, d_{n-1}, d_n$ e $/d_1d_2/d_3, \dots, d_{n-1}, d_n$ são estados ruins, caso contrário, $d_2/d_1/d_3, \dots, d_{n-1}, d_n$ e $d_2, d_1//d_3, \dots, d_{n-1}, d_n$ são estados ruins. Em ambos os casos, é possível alcançar o estado $d_1, d_2, \dots, d_{n-1} // d_n$.

Portanto, todos os estados ruins nos quais $d_n \in p_2$ estão conectados. Com um

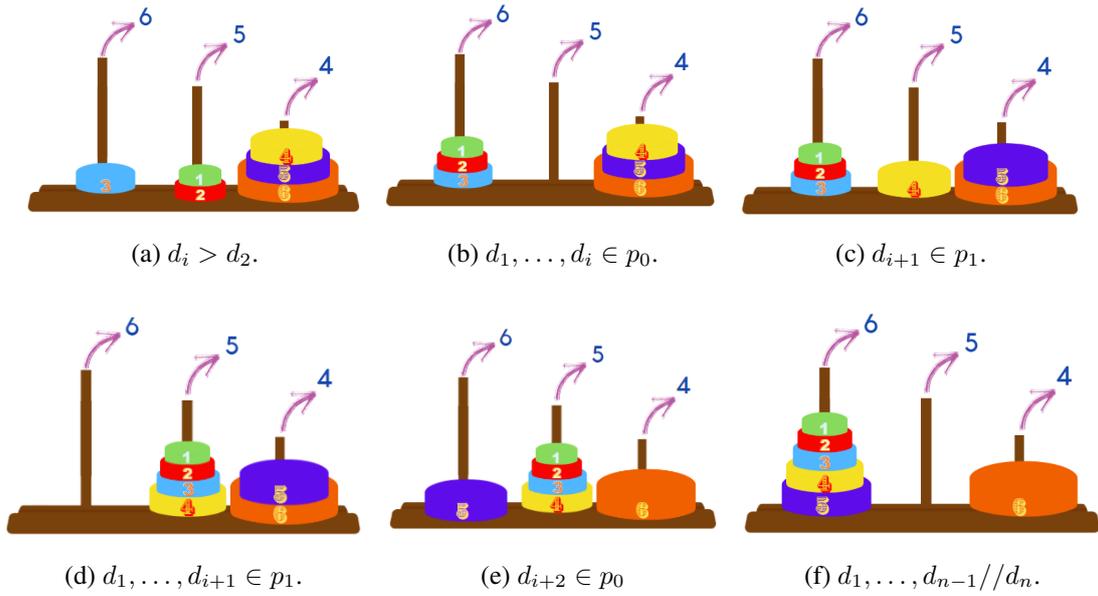


Figura 5.6. ToM: exemplo de uma sequência de movimentos para ir de um estado ruim em que d_n está em p_2 para o estado $d_1, d_2, \dots, d_{n-1} // d_n$.

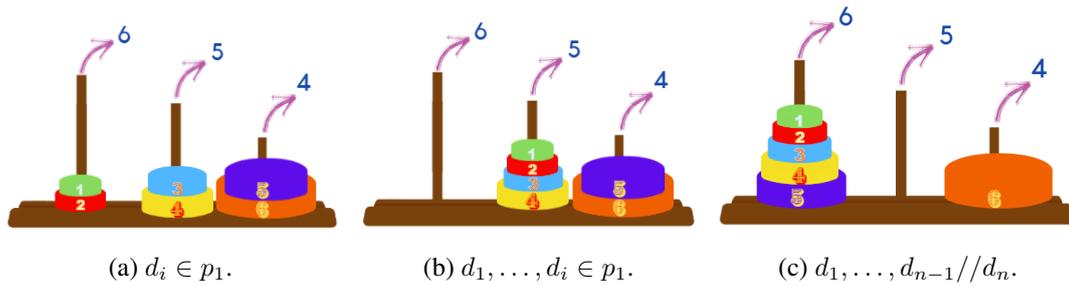


Figura 5.7. ToM: outra sequência de movimentos para ir de um estado ruim em que d_n está em p_2 para o estado $d_1, d_2, \dots, d_{n-1} // d_n$.

procedimento semelhante, pode-se transicionar de qualquer estado em que $d_n \in p_1$ e chegar no estado $d_1, d_2, \dots, d_{n-1} // d_n /$, que por sua vez é adjacente a $d_1, d_2, \dots, d_{n-1} // d_n$. Isso implica que todos os estados ruins estão conectados também.

□

Dado que o grafo da ToM O_3^n possui uma componente com os vértices bons e outra com os vértices ruins, surge o Teorema 6 para demonstrar os casos em que esses vértices são bons ou ruins.

Teorema 6 (Complementaridade de estados bons e ruins). *Na ToM um estado $X/Y/Z$ é um estado bom do jogo se, e somente se, $Y/X/Z$ for um estado ruim, exceto o estado $d_1, d_2, \dots, d_n //$, para todo $n \geq 2$.*

Demonstração. Seja $s = X/Y/Z$ um estado bom do jogo da ToM diferente de $d_1, d_2, \dots, d_n//$. Como, pelos Teoremas 2 e 3, todo estado em que $d_n \in p_0$ é bom e todo estado em que $d_n \in p_1$ é ruim, então, se d_n está em p_0 ou p_1 , o resultado é trivialmente válido. Agora, precisa-se verificar os casos em que d_n está em p_2 . Seja $s' = Y/X/Z$. Observe que em s , há no máximo $n - 1$ discos em p_0 , então s' não é um estado irregular.

Primeiramente, mostra-se que não é possível alcançar s' a partir de s . Suponha, por contradição, que exista uma sequência de movimentos para ir de s para s' e considere uma sequência mínima. Seja d_i o maior disco fora de p_2 em s , lembre-se de que $i \leq n - 1$. No caso em que d_i está em p_0 , considere a primeira vez que d_i é movido nesta sequência. Antes disso, no máximo um disco maior que d_i poderia ter sido movido, porque p_0 não estava disponível para discos maiores que d_i . Se d_i é movido para p_1 , então, p_1 estava vazio ou d_{i+1} é o único disco em p_1 .

O primeiro caso é impossível, pois exigiria $n - 1$ discos em p_2 . O último é uma contradição com o fato de que essa é uma sequência mínima de movimentos, porque mover d_{i+1} de volta para p_2 exigiria mover d_i de volta para p_0 com os mesmos discos sobre ele. Assim, d_i foi movido para p_2 e os discos d_1, d_2, \dots, d_{i-1} estão todos em p_1 , enquanto d_i, d_{i+1}, \dots, d_n estão em p_2 . A partir desse ponto, é necessário mover todos os discos de p_1 para p_0 antes de mover d_i para p_1 . No entanto, isso não é possível, porque, para mover d_{i-1} para p_0 , todos os outros discos devem estar em p_2 , violando sua capacidade.

O caso em que d_i está em p_1 é simétrico a quando está em p_2 . Portanto, não é possível alcançar s' a partir de s e vice-versa. Como os estados bons são conectados por definição, s e s' não podem ser ambos bons, nem ambos ruins, porque, pelo Teorema 5, os estados ruins também estão conectados. Assim, se s é bom, então s' deve ser ruim e vice-versa, comprovando a complementaridade de estados bons e ruins. \square

Como o M_3^n é um subgrafo de H_3^n , é possível analisar o primeiro com base na estrutura do segundo. H_3^n tem 3 cópias de H_3^{n-1} disjuntas, que serão denotadas por TOP, ESQ, DIR , segundo o esquema demonstrado na Figura 5.8. Em geral, H_3^n pode ser representado como uma estrutura triangular com os vértices de grau 2 nos vértices desse triângulo 3. Considerando essa representação, um vértice de grau 2 é denotado por

³Existem apenas 3 vértices de grau 2, ver na Seção 3.4, Proposição 1.

extremidade, que mais especificamente pode ser a **extremidade do topo, da direita ou da esquerda**. Aplicando essa estrutura ao grafo da Torre de Manaus, é possível demonstrar o Lema 4.

Lema 4 (Estrutura recursiva do grafo M_3^n). *Particionando M_3^n nos subgrafos TOP , ESQ , DIR seguindo o esquema recursivo de H_3^n , todas as seguintes assertivas são válidas:*

- i) TOP é isomorfo a H_3^{n-1} menos a extremidade da esquerda;
- ii) ESQ é isomorfo a H_3^{n-1} menos as extremidades do topo e da esquerda;
- iii) DIR é isomorfo a M_3^{n-1} adicionando-se a extremidade da esquerda.

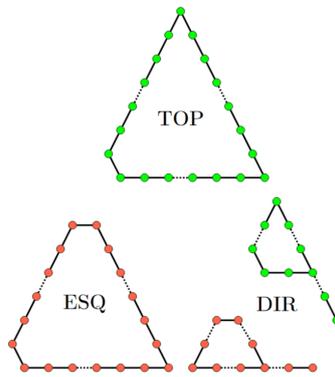


Figura 5.8. Estrutura do grafo da ToM, onde: TOP - representa o topo; ESQ - o lado esquerdo; e, DIR - o lado direito do grafo.

Demonstração. Abaixo são explicados os três casos ilustrados na Figura 5.8:

- i) Considerando H_3^n , é possível observar que TOP é o subgrafo contendo os estados em que $d_n \in p_0$. Com exceção do estado $d_n/1, 2, \dots, d_{n-1}$, todos esses estados são válidos no jogo ToM, logo TOP tem um vértice a menos que H_3^{n-1} que é justamente a extremidade da esquerda que denota o estado $d_n/1, 2, \dots, d_{n-1}$;
- ii) Considerando H_3^n , é possível ver que ESQ é o subgrafo contendo os estados em que $d_n \in p_1$. Com exceção dos estados $d_n/1, 2, \dots, d_{n-1}$ e $/1, 2, \dots, d_n/$, em todos os outros estados os pinos p_1, p_2 e p_3 não excedem suas capacidades máximas. Logo, ESQ tem dois vértices a menos que H_3^{n-1} que são exatamente as extremidades: do topo que denota o estado $d_n/1, 2, \dots, d_{n-1}$ e a extremidade da esquerda que denota o estado $/1, 2, \dots, d_n/$;

iii) Como DIR é o subgrafo que contém os estados em que $d_n \in p_2$, tem-se que o estado $X/Y/Z$ é um vértice de DIR , se e somente se, o vértice $Y/X/Z$ também pertence a DIR , pois essa inversão não muda d_n de pino. Aplicando o Teorema 6 nos vértices de DIR , pode-se concluir que a estrutura do grafo da ToM se repete nesse subgrafo. Mas, o Teorema 6 excetua o vértice $d_1, d_2, \dots, d_{n-1} // d_n$, que é a extremidade do topo. No entanto, $d_1, d_2, \dots, d_{n-1} // d_n$ é um estado válido de M_3^n . Logo, DIR tem os mesmos vértices que M_3^{n-1} mais o vértice $/d_1, d_2, \dots, d_{n-1} // d_n$ que é a extremidade da esquerda.

□

Um exemplo da aplicação da estrutura do Lema 4 no grafo M_3^4 pode ser visto na Figura 5.9, em que se observa claramente que TOP é um H_3^3 menos o vértice da extremidade esquerda $4//1, 2, 3$. ESQ é um H_3^3 menos os dois vértices das extremidades topo e esquerda, $/4/1, 2, 3$ e $/1, 2, 3, 4/$, respectivamente. Com isso, DIR é M_3^3 mais um vértice na extremidade da esquerda $1, 2, 3//4$.

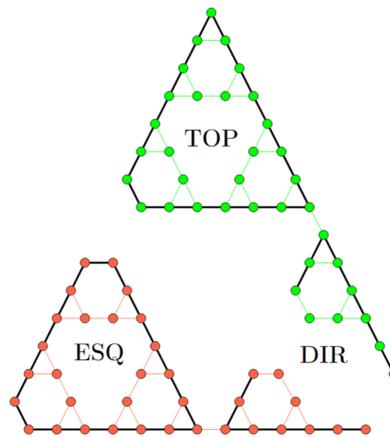


Figura 5.9. Exemplo da estrutura de particionamento do Lema 4 no grafo M_3^4 .

Os movimentos necessários para solucionar o jogo com 3 discos, tendo o estado regular inicial $1/2/3$, são apresentados na Figura 5.10. O caminho destacado em amarelo, são os vértices a serem percorridos, demonstrando que foram necessários 5 movimentos para resolver o jogo. Isto está de acordo com o que foi apresentado na Figura 5.2, sendo o número mínimo de movimentos necessários para solucionar o jogo. No grafo M_3^3 esse vértice é o que requer o maior número de movimentos para chegar a solução do jogo, como demonstrado no Teorema 7.

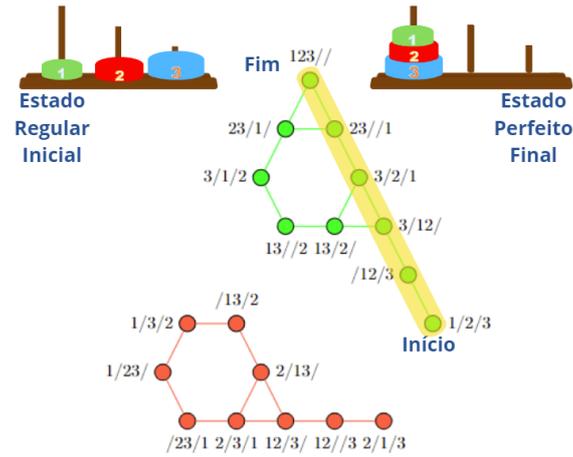


Figura 5.10. M_3^3 : em amarelo o caminho ótimo a ser percorrido para solucionar o jogo, partindo do estado inicial 1/2/3.

Teorema 7 (Análise de pior caso do jogo da ToM). *No pior caso são necessários $2^n - 3$ movimentos para resolver o jogo da ToM, quando $n \geq 2$.*

Demonstração. A prova será por contradição tratando-se 2 casos, supondo que no 1º caso existe um estado s cuja solução ótima requer um número maior ou igual que $2^n - 3$ movimentos para resolver o jogo da ToM e no 2º caso, se existe outro estado diferente de s que precise de um número menor ou igual a $2^n - 3$ movimentos para resolver o jogo da ToM:

No 1º caso, tendo como base o menor número de movimentos do jogo da ToH, $2^n - 1$, pois, M_3^n é um subgrafo de H_3^n . Neste último grafo, na extremidade da direita, tem-se o caminho ótimo onde todos os d_1, d_2, \dots, d_n discos são movidos até estarem todos em p_2 . Porém, na ToM, p_2 só comporta $n - 2$ discos. Por isso, sempre existirão 2 vértices do caminho ótimo do grafo da ToH que não poderão existir no grafo da ToM. Dependendo da paridade de n , esses dois vértices mudam. Quando n for par, não existirão os vértices $/1/d_2, \dots, d_n$ e $//d_1, d_2, \dots, d_n$. Quando n for ímpar, não existirão os vértices $1//d_2, \dots, d_n$ e $//d_1, d_2, \dots, d_n$. Logo, o maior caminho que existirá no grafo M_3^n será $2^n - 1 - 2$ vértices, ou seja, $2^n - 3$.

Do estado s é preciso passar por $2^n - 3$ vértices no grafo para chegar ao estado perfeito final do jogo. Quando a quantidade de discos em s for par, o estado s será $2/1/d_3, \dots, d_n$. Quando a quantidade de discos for ímpar, s será $1/2/d_3, \dots, d_n$. Partindo de s , executando o algoritmo iterativo da ToH, Algoritmo 2 de forma reversa, ou seja, de

s para o estado final, serão necessários $2^n - 3$ movimentos.

No 2º caso, por contradição, supondo que exista um outro vértice (representando um outro estado) que a solução exija mais que $2^n - 3$ movimentos, com base na estrutura proposta no Lema 4, tal vértice não poderia estar em ESQ , porque em ESQ todos os vértices representam estados ruins. Se esse outro vértice estivesse em TOP , no máximo ele teria $2^{n-1} - 1$ movimentos, pois, como foi visto no Lema 4, TOP é o subgrafo H_3^{n-1} . Logo, esse vértice só pode estar em DIR , porém, o vértice mais distante em DIR é justamente o vértice s , como foi provado no 1º caso. Então, é uma contradição que exista um outro vértice em leve a uma solução que seja necessário mais que $2^n - 3$ movimentos.

Portanto, o pior caso na ToM é $2^n - 3$ movimentos. \square

Com isso, o jogo da Torre de Manaus, assim como da Torre de Hanoi, também é exponencial em discos, tendo $O(2^n)$. Ainda comparando com o grafo H_3^n o Teorema 8 prova que o número de vértices do grafo da ToM é $3^n(n + 1)$.

Teorema 8 (Total de vértices do M_3^n). *O grafo do jogo da Torre de Manaus M_3^n possui $3^n - 2(n + 1)$ vértices, quando $n \geq 2$.*

Demonstração. Tendo como base o grafo da ToH - H_3^n , o qual comprovadamente possui 3^n vértices e obedece às duas primeiras regras do jogo da ToM, lembrando: somente pode mover um disco de cada vez; e, a segunda, um disco maior não pode ser colocado em cima de outro menor. O foco deve a ser a terceira regra do jogo: as capacidades máximas dos pinos do jogo são da esquerda para a direita, n , $n - 1$ e $n - 2$ discos. Com essa regra, vários estados que na ToH eram regulares, passam a ser irregulares na ToM.

Nesse cenário, averiguando o jogo para descobrir quantos estados irregulares ocorrem, existem 3 casos:

No 1º caso: verifica-se quantos estados são irregulares quando todos os discos estão em p_0 , todos os estados são possíveis, porque ele suporta os n discos;

No 2º caso: analisa-se quando todos os discos estão em p_1 , como ele permite somente $n - 1$ discos, tem 1 estado irregular, onde todos os n discos estão nesse pino;

No 3º caso: examina-se quando todos os discos estão em p_2 , primeiramente ocorre a mesma situação do 2º caso, pois, ele não comporta todos os n discos, então 1 estado

irregular, e nem os $n - 1$ discos. Porém nessa situação, têm-se n possibilidades de ter 1 disco fora de p_2 , mas esse disco pode estar em p_0 ou em p_1 , logo têm-se $2n$ estados irregulares.

Portanto, o total de estados irregulares é $1 + 1 + 2n = 2(n + 1)$. Com base nisto, para calcular o número de vértices no grafo M_3^n basta retirar os estados irregulares (vértices) do total de vértices da ToH. Sendo assim,

$$\begin{aligned} |V(M_3^n)| &= |V(H_3^n)| - 2(n + 1) \\ |V(M_3^n)| &= 3^n - 2(n + 1) \end{aligned}$$

□

Como o jogo da ToM possui os estados regulares divididos em bons ou ruins, o Teorema 9 prova por indução o número de vértices bons.

Teorema 9 (Quantidade de vértices bons no M_3^n). *O grafo M_3^n possui $\lfloor \frac{3^n}{2} \rfloor - n$ vértices bons, quando $n \geq 2$.*

Demonstração. Por indução em n .

Caso Base (CB): $n = 2$.

$$\begin{aligned} |V_b(M_3^2)| &= \lfloor \frac{3^2}{2} \rfloor - 2 \\ |V_b(M_3^2)| &= 2 \end{aligned}$$

O que pode ser comprovado vendo M_3^2 , Figura 5.4a, em que mostra que M_3^2 possui somente 2 vértices bons quando o jogo possui somente 2 discos.

Hipótese de Indução (HI): O grafo M_3^k possui $\lfloor \frac{3^k}{2} \rfloor - k$ vértices bons, para $k \geq 2$.

Passo Indutivo (PI): Seja um grafo $M_3^{k+1} = (V_b \cup V_r, E)$, um grafo da ToM com $k + 1$ discos.

Focando nos vértices bons do grafo M_3^{k+1} , pela contra positiva do Teorema 4, o jogo só tem solução quando o maior disco do jogo estiver em p_0 (1º caso) ou em p_2 (2º caso).

No 1º caso, considerando a ToM com 3 pinos e $k+1$ discos (Figura 5.11a), quando o disco maior d_{k+1} está no p_0 , sobram k discos e têm-se os mesmos estados(vértices) que a ToH com k discos, $|V(H_3^k)| = 3^k$, menos um vértice, que seria o estado em que todos os discos estariam em p_2 . Como a capacidade de p_2 é $k - 1$, não é possível colocar os k discos neste pino. Logo, com d_{k+1} em p_0 existem $3^k - 1$ estados bons;

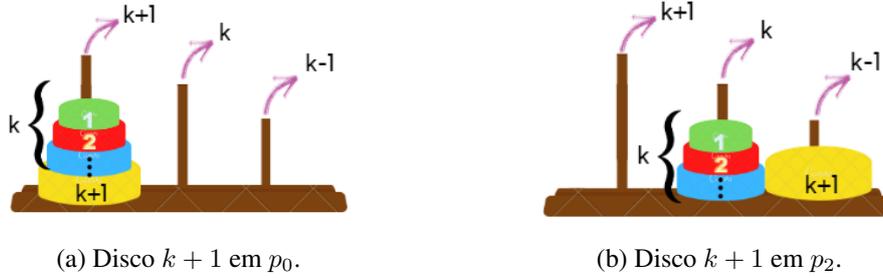


Figura 5.11. Torre de Manaus com $k + 1$ discos.

No 2º caso, quando o disco maior d_{k+1} está em p_2 , será necessário mover todos os k discos para p_1 (Figura 5.11b) e em seguida o $k + 1$ para p_0 . Entretanto, considerando as restrições de capacidade dos pinos, isso nem sempre é possível. Em particular, se d_k estiver em p_0 , não terá solução. Em geral, a solução de ToM, quando d_{k+1} estiver em p_2 , depende da solução da ToM com k discos. Pela HI, o grafo da ToM possui $[\lfloor \frac{3^k}{2} \rfloor - k]$ vértices bons.

Desse modo, no total têm-se:

$$\begin{aligned}
 |V_b(M_3^{k+1})| &= (3^k - 1) + (\lfloor \frac{3^k}{2} \rfloor - k) \\
 |V_b(M_3^{k+1})| &= (3^k - 1) + (\frac{3^k - 1}{2} - k) \\
 |V_b(M_3^{k+1})| &= \frac{2 \cdot 3^k - 2 + 3^k - 1 - 2k}{2} \\
 |V_b(M_3^{k+1})| &= \frac{3 \cdot 3^k - 3 - 2k}{2} \\
 |V_b(M_3^{k+1})| &= \frac{3^{k+1} - 1 - 2(k+1)}{2} \\
 |V_b(M_3^{k+1})| &= \frac{3^{k+1} - 1}{2} - (k + 1) \\
 |V_b(M_3^{k+1})| &= \lfloor \frac{3^{k+1}}{2} \rfloor - (k + 1)
 \end{aligned}$$

Logo, o grafo M_3^n possui $\lfloor \frac{3^n}{2} \rfloor - n$ vértices bons. □

Analisando o total de vértices do grafo do jogo da TOM, retirando o número de vértices bons, conclui-se o número de vértices ruins definido no Corolário 2.

Corolário 2 (Quantidade de vértices ruins no M_3^n). *O grafo M_3^n possui $\lfloor \frac{3^n}{2} \rfloor - (n + 1)$ vértices ruins, quando $n \geq 2$.*

Portanto, nesta seção foram relacionadas as propriedades do grafo do jogo da Torre de Manaus, sendo definido que ele sempre terá duas componentes e que no pior caso serão necessários $2^n - 3$ movimentos para chegar ao vértice que representa o estado perfeito final do jogo. Também foi expresso o esquema recursivo para gerar a estrutura desse grafo, o número total de vértices, quantos desses vértices são bons e, por fim, foi quantos vértices são ruins.

5.4. Aplicações da Torre de Manaus

O jogo da Torre de Manaus pode ser aplicado no processo de ensino e aprendizagem de conteúdos de computação, como de matemática. Assim como, em muitas situações reais, na ToH, pode-se citar os problemas de pilha na gestão de logística, relatado na Seção [D](#) e os problemas de *Backup* na área de TI, descrito na Seção [E](#). Logo, esse jogo também herda as aplicações da ToL, podendo também, ser aplicado na psicologia para avaliar a capacidade do indivíduo de planejar e solucionar problemas.

Portanto, todas as aplicações abordadas no decorrer deste trabalho, também se moldam para a Torre de Manaus, acrescentando ainda mais dificuldade dada à restrição de capacidade das torres e a questão de poder incluir casos, estados e/ou estruturas proibidas. O que não acontece nas demais variações da ToH já existentes.

5.5. Trabalhos Relacionados

Os trabalhos relacionados ao Jogo da Torre de Manaus são todos os que envolvem a proposição de um jogo combinatório, baseado em torres e que explore aspectos matemático-computacionais, envolvendo modelagem teórica, principalmente, em grafos e aspectos de complexidade computacional e algoritmos.

No decorrer deste trabalho, o Capítulo [3](#) explana sobre o Jogo da Torre de Hanoi [\[Hinz, Klavžar e Petr 2018\]](#) e o Capítulo [4](#), que na Seção [4.2](#) descreve o jogo da ToB [\[Herter e Rote 2018\]](#), na Seção [4.3](#) que demonstra o teste da ToL [\[Berg e Byrd 2002\]](#) e por fim, a Seção [4.4](#) que exhibe o jogo da ToO [\[Unterrainer et al. 2005\]](#). Todos esses jogos de torres tratam sobre aspectos relacionados ao jogo em proposição. Adicionalmente, no Capítulo [6](#) a seguir, na Seção [6.3](#), é realizada uma análise comparativa sobre os grafos das variações exploradas.

Capítulo 6

Outros Resultados

Neste capítulo, é abordada a generalização do grafo da Torre de Hanoi por estruturas em grafos, utilizando a operação de enxerto para construí-lo. Além disso, são mostradas suas semelhanças com o cobra triangular alternada. Assim como são descritas as contribuições adicionadas em jogos de torres já criados e encontrados na literatura, realizando refinamentos que transformam testes neurocognitivos, como no caso da Torre de Londres, em um jogo combinatório. Finalizando, será apresentada uma análise comparativa entre a ToH e as variações: ToB, ToO, ToL e ToM.

6.1. Modelagem e Generalização do Grafo da Torre de Hanoi por Estruturas em Grafos

Estudando a composição do H_3^n , observou-se várias estruturas de grafos simples que se repetem e uma grande semelhança deste com o cobra triangular alterna - $A(T_6)$, abordadas na Subseção [6.1.3](#), o qual serviu de *insight* para uma nova abordagem quanto à sua construção e que será detalhado na Subseção [6.1.2](#).

Analisando a parte interna do grafo, encontram-se vários subgrafos elementares em formato de triângulos - C_3 , Figura [6.1a](#) e têm-se ainda várias figuras hexagonais, ou seja, subgrafos elementares - C_6 como na Figura [6.1c](#). Explorando externamente o grafo, percebe-se a presença de outras estruturas triangulares formadas por 9 vértices, Figura [6.1b](#).

Reagindo a essa observação, surgiu a ideia de generalização da construção do grafo da ToH [[Martins, Costa e Freitas 2024](#)] por meio de um processo aqui denominado

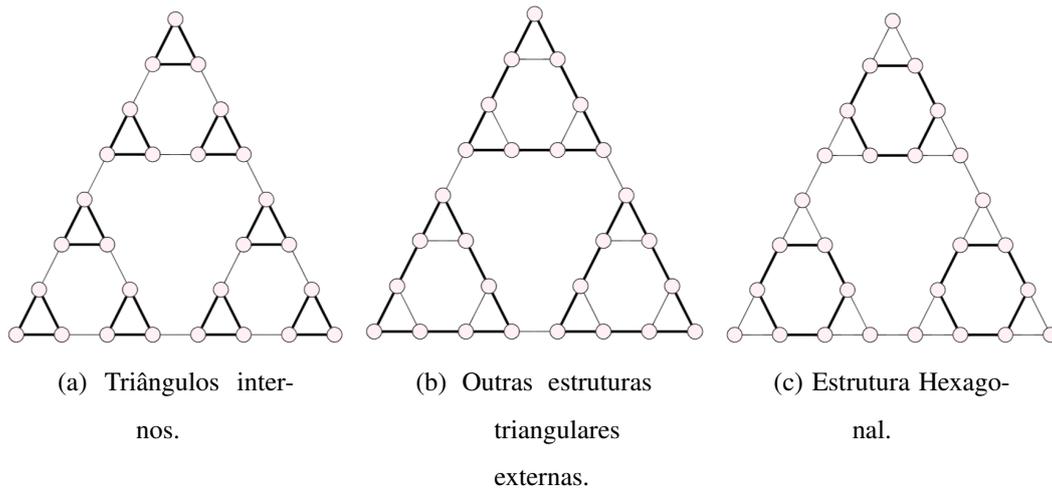


Figura 6.1. Grafo H_3^3 destacando as estruturas (a) C_3 , (b) C_9 e (c) C_6 .

de operação de enxerto, formalmente descrita na Subseção 6.1.1. Na Subseção 6.1.2, é apresentado um algoritmo que generaliza a construção do grafo da ToH para n discos.

6.1.1. Operação de Enxerto

Dado um grafo G , com pelo menos uma aresta ab , como apresentado na Figura 6.2a, bem como, um grafo Q qualquer, com pelo menos dois vértices v, w , como apresentado na Figura 6.2b. A operação de enxerto $enx(Q, v, w, G, ab)$ é um tipo de união dos grafo G e Q e é dada pela remoção da aresta ab de G , seguida da união adjunta de G e Q e identificação dos vértices “ a e v ”, e “ b e w ”, como apresentado na Figura 6.3 e na Definição 6.

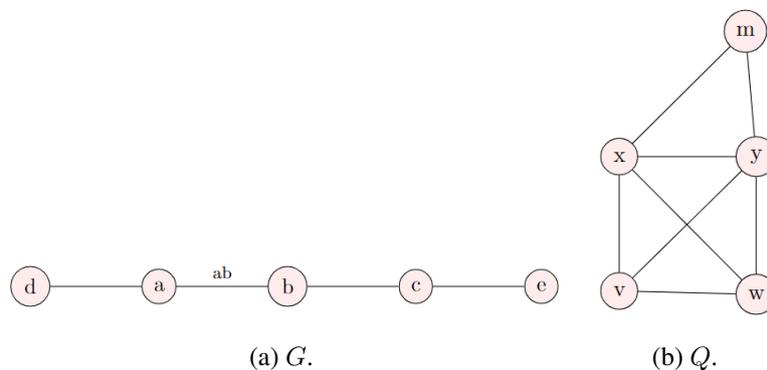


Figura 6.2. Grafos (a) G e (b) Q para exemplificar a operação de enxerto.

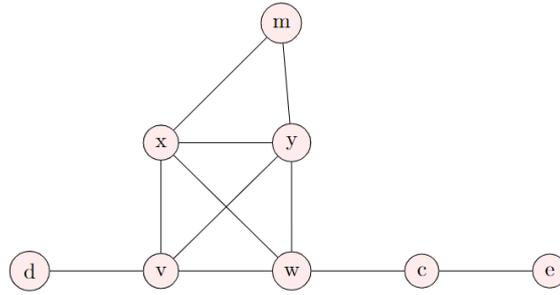


Figura 6.3. Grafo R , resultante da operação de enxerto, $R = \text{enx}(Q, v, w, G, ab)$, do grafo Q , Figura 6.2b, no grafo G , Figura 6.2a.

Definição 6 (Operação de enxerto). A operação de enxerto resulta no grafo $R = \text{enx}(Q, v, w, G, ab)$, tal que $V(R) = V(G) \setminus \{a, b\} \cup V(Q)$ e $E(R) = E(Q) \cup \{vz \mid z \in V(G), az \in E(G), z \neq b\} \cup \{wp \mid p \in V(G), bp \in E(G), p \neq a\} \cup \{zp \mid z, p \notin \{a, b\}, zp \in E(G)\}$.

Seja $R = \text{enx}(Q, v, w, G, ab)$, note que a aresta ab só existirá em $E(R)$ se, somente se, $vw \in E(Q)$, ou seja, $ab \in E(R) \leftrightarrow vw \in E(Q)$. Exemplo: Dado um grafo G' , contendo uma aresta ab , como apresentado na Figura 6.4a, e um grafo Q' contendo os vértices v e w , como na Figura 6.4b, ao realizar-se a operação de enxerto entre os grafos G' e Q' obtem-se o grafo $R' = \text{enx}(Q', v, w, G', ab)$, como mostra a Figura 6.4c.

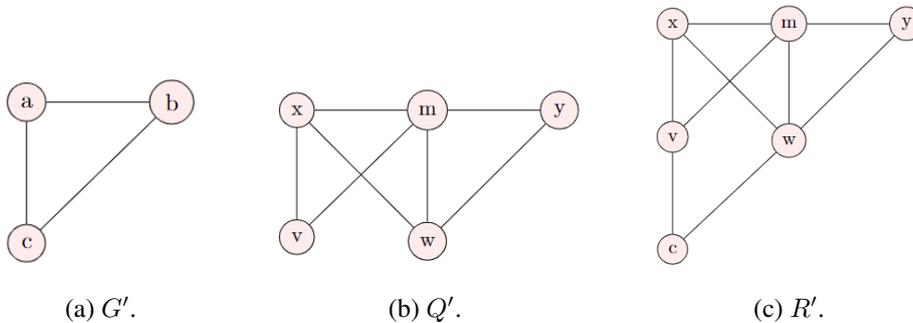


Figura 6.4. Operação de enxerto $R' = \text{enx}(Q', v, w, G', ab)$, explicitando que a aresta ab somente existirá em $E(R)$ se, somente se, $vw \in E(Q)$.

Portanto, observa-se que após a operação $\text{enx}(Q', v, w, G', ab)$ ser realizada, a aresta ab somente existirá em $E(R)$ se, somente se, $vw \in E(Q)$.

Além disso, para enxertar múltiplas cópias de um grafo Q em G , por conveniência, utiliza-se o seguinte abuso de notação: $\text{enx}(Q, v, w, G, M)$, onde M é um emparelhamento qualquer de G . O que pode ser definido indutivamente como: $\text{enx}(Q, v, w, G, \{ab\}) = \text{enx}(Q, v, w, G, ab)$ e $\text{enx}(Q, v, w, G, M) =$

$\text{enx}(Q', v', w', \text{enx}(Q, v, w, G, ab), M \setminus \{ab\})$, em que Q' é uma cópia de Q e ab é uma aresta qualquer de M . Esse mesmo processo também pode ser realizado de maneira iterativa, como no Algoritmo [13](#).

Algoritmo 13: Enxerto de múltiplas cópias de Q	
Entrada: Um grafo Q , dois vértices $v, w \in V(Q)$, grafo G e um emparelhamento M de G .	
Saída: O grafo $\text{enx}(Q, v, w, G, M)$	
1	enx (Q, v, w, G, M)
2	$R \leftarrow G$;
3	para $ab \in M$ faça
4	$Q' \leftarrow$ cópia de Q ;
5	$R \leftarrow$ enx (Q', v', w', R, ab)
6	$M \leftarrow M \setminus \{ab\}$;
7	fim
8	retorna R
9	fim

Observe que, uma vez que as arestas de M formam um emparelhamento, sempre será possível realizar todos os $|M|$ enxertos e em qualquer ordem.

6.1.2. Construção do Grafo da Torre de Hanoi usando Enxertos em Ciclos C_6

O grafo da ToH para n discos pode ser gerado enxertando 3 cópias de H_3^{n-1} em um emparelhamento perfeito de um C_6 . Isto é, $H_3^n = \text{enx}(H_3^{n-1}, v, w, C_6, M)$, em que v, w são dois vértices de grau 2 de H_3^{n-1} e M é um emparelhamento perfeito de C_6 , conforme descrito no Algoritmo [14](#) e ilustrado na Figura [6.5](#).

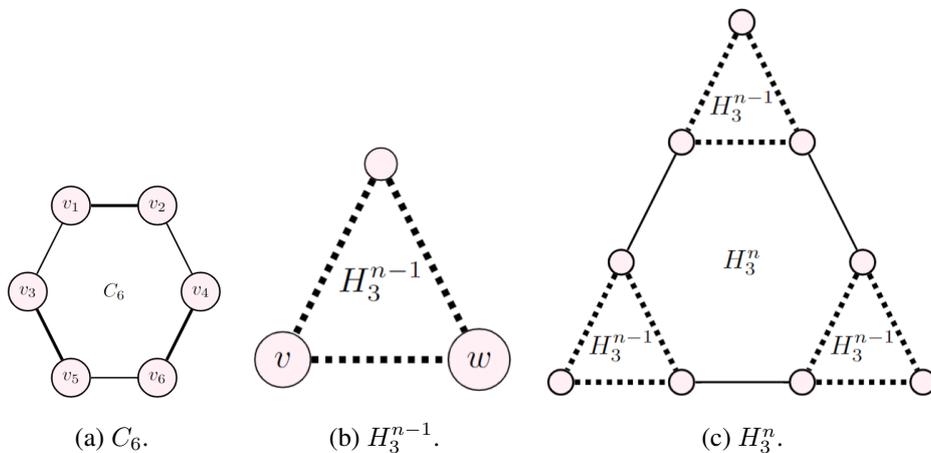


Figura 6.5. Ilustração do processo recursivo de construção do Grafo H_3^n (c), onde (a) C_6 destacando as arestas pertencentes ao emparelhamento perfeito, (b) o grafo H_3^{n-1} que será enxertado.

Algoritmo 14: Construção do grafo H_3^n .**Entrada:** o número de discos $n, n \geq 1$.**Saída:** Grafo H_3^n .

```

1 ConstruaGrafoToH ( $n$ )
2   se ( $n == 1$ ) então
3     retorna  $H_3^1$ 
4   senão
5      $H_3^{(n-1)} \leftarrow$  ConstruaGrafoToH ( $n - 1$ )
6      $v, w \leftarrow$  dois vértices de grau 2 de  $H_3^{(n-1)}$ 
7      $M \leftarrow$  emparelhamentoMaximo( $C_6$ )
8      $H_3^n \leftarrow$  enx( $H_3^{n-1}, v, w, C_6, M$ )
9     retorna  $H_3^n$ 
10  fim
11 fim

```

Observa-se o algoritmo [14] é recursivo, tendo como caso base o grafo com apenas 1 disco, H_3^1 . E, a partir de então, com a operação de enxerto, pode-se gerar qualquer grafo H_3^n , em que é necessário que o grafo Q seja H_3^{n-1} . Desse modo, as Figuras [6.6a], [6.6b] e [6.6c] representam o processo do Algoritmo [14] para a geração do grafo H_3^3 , a partir do enxerto de 3 cópias do grafo H_3^2 no grafo C_6 .

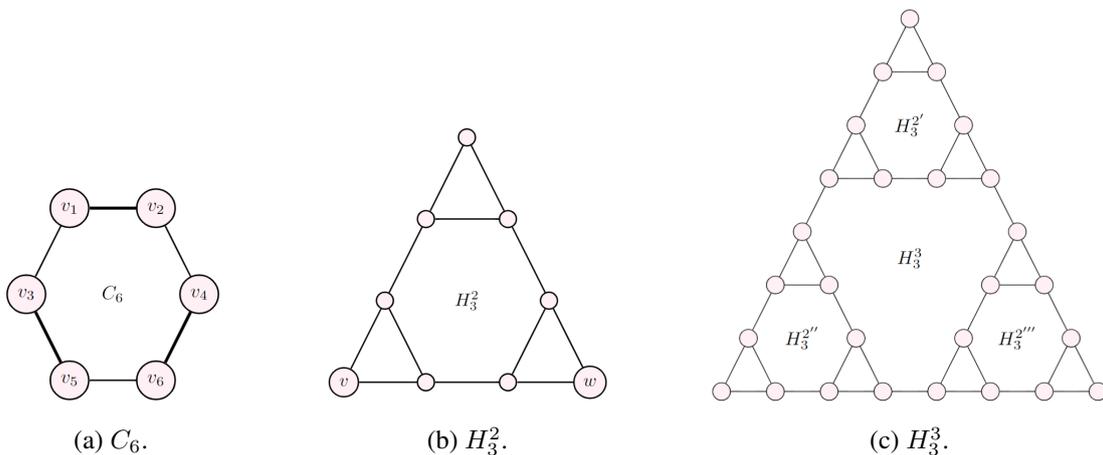


Figura 6.6. Exemplo do processo recursivo de construção do Grafo H_3^3 (c), onde (a) C_6 em destaque as arestas pertencentes ao emparelhamento perfeito, (b) o grafo H_3^2 que foi enxertado nas arestas do C_6 .

Adicionalmente, essa definição recursiva do grafo da Torre de Hanoi ajuda a comprovar outras propriedades como aponta o Teorema [10].

Teorema 10 (Quantidade de C_3 no H_3^n). *O grafo da Torre de Hanoi H_3^n possui $3^{n-1} C_3$, quando $n \geq 1$.*

Demonstração. Por indução em n .

CB: $n = 1$

$$3^{1-1} = 1 C_3$$

Isso pode ser confirmado, retornando à Seção 3.4, verificando o grafo H_3^1 na Figura 3.3b.

HI: H_3^k tem $3^{(k-1)}C_3$ disjuntos, para algum $k \geq 1$.

PI: Seja um grafo H_3^{k+1} , um grafo da ToH com $k + 1$ discos.

Por construção H_3^{k+1} , Figura 6.5c, possui $3H_3^k$ disjuntos em vértices (e mais 3 arestas que os conectam). Enquanto que, por HI, cada um deles tem $3^{(k-1)}C_3$ disjuntos. Logo, H_3^{k+1} tem $3(3^{(k-1)}) = 3^k C_3$ disjuntos, como deseja-se provar. \square

6.1.3. Similaridades do Grafo da Torre de Hanoi e o Cobra Triangular Alternada

O grafo da Torre de Hanoi com 2 discos pode ser desenhado baseado no Grafo Cobra Triangular Alternada $A(T_6)$, unindo, portanto, 3 cópias disjuntas H_3^1 , $H_3^{1'}$ e $H_3^{1''}$ de H_3^1 . Para completar o grafo H_3^2 , adiciona-se apenas uma aresta para ligar os vértices u_1 a u_6 , como apresentado na Figura 6.7.

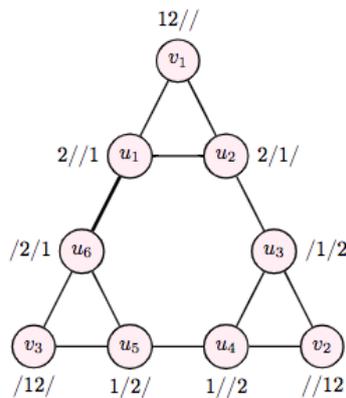


Figura 6.7. Grafo H_3^2 baseado no Cobra Triangular Alternada $A(T_6)$, em destaque a aresta u_1u_6 que foi adicionada para completar o grafo.

Igualmente, para gerar H_3^3 , pegam-se 3 cópias disjuntas $H_3^{2'}$, $H_3^{2''}$ e $H_3^{2'''}$ de H_3^2 , rotulando cada C_3 como $u'_1, u'_2, u'_3, u'_4, u'_5, u'_6, v'_1, v'_2$ e v'_3 e adicionam-se 3 arestas para

ligar as cópias de H_3^2 , como na Figura 6.8. O grafo resultante corresponde ao grafo $A(T_6)$, tendo $9C_3 + 12$ arestas.

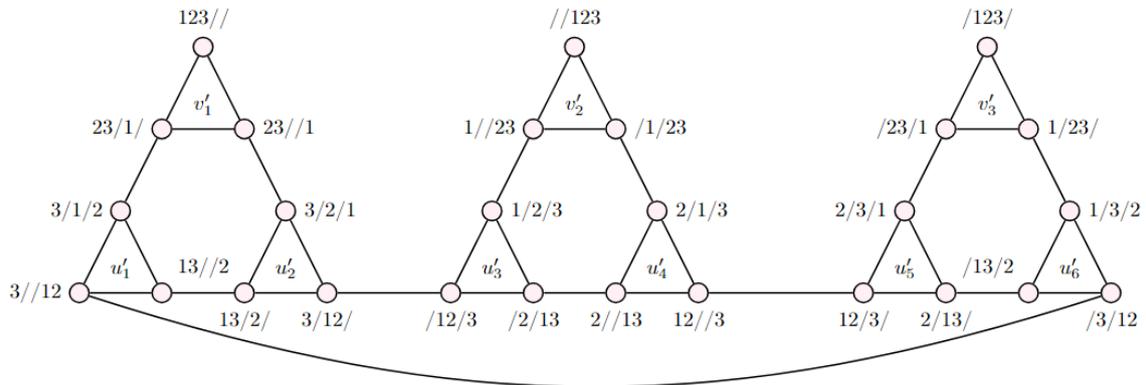


Figura 6.8. H_3^3 baseado no Cobra Triangular Alternada $A(T_6)$.

Assumindo que H_3^2 é um $A(T_6)$. Para esboçar H_3^3 , pegam-se 3 cópias disjuntas $H_3^{2'}$, $H_3^{2''}$ e $H_3^{2'''}$ de H_3^2 e adicionam-se 3 arestas para estabelecer a ligação com todos os $A(T_6)$, como apresentado na Figura 6.8. Assim, H_3^3 possui $3A(T_6)$ ligados por 3 arestas. Logo, para $n = 4$, H_3^4 é composto por $9A(T_6)$ com 12 arestas (Figura 6.9), como definido no Teorema 11.

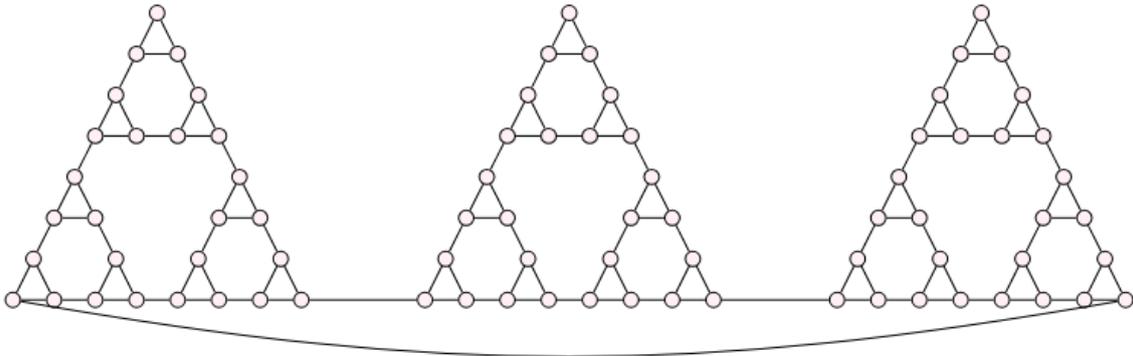


Figura 6.9. H_3^4 baseado no Cobra Triangular Alternada $A(T_6)$.

Teorema 11 (Quantidade de $A(T_6)$ no H_3^n). *O grafo da Torre de Hanoi H_3^n tem $3^{(n-2)}$ $A(T_6)$, quando $n \geq 2$.*

Demonstração. Por indução em n .

CB: $n = 2$

$$3^{(2-2)} = 1 A(T_6)$$

Isso pode ser confirmado, visualizando o grafo H_3^2 na Figura 6.7.

HI: H_3^k tem $3^{(k-2)}$ $A(T_6)$ disjuntos, para algum $k \geq 2$.

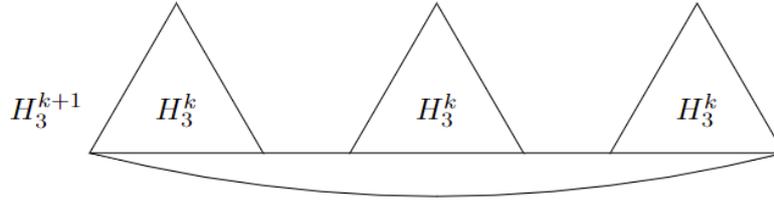


Figura 6.10. Grafo H_3^{k+1} composto por $3(H_3^k)$ disjuntos em vértices.

PI: Seja um grafo H_3^{k+1} , um grafo da ToH com $k + 1$ discos.

Por construção H_3^{k+1} , Figura 6.10, possui $3H_3^k$ disjuntos em vértices (e mais 3 arestas que os conectam, mas sem formar um novo $A(T_6)$). Por HI cada um deles tem $3^{(k-2)}$ $A(T_6)$ disjuntos. Logo, H_3^{k+1} tem $3 \times (3^{k-2}) = 3^{k-1}$ $A(T_6)$ disjuntos, como desejava-se demonstrar. \square

Assumindo que H_3^{n-1} tem cada ciclo C_3 formado por triângulos, que são compostos por 3 vértices. Então, o grafo H_3^3 possui $9C_3$, o que é expresso pelo Teorema 10.

Nesta seção definiu-se a operação de enxerto, a qual foi utilizada para generalizar a construção do grafo da ToH somente com estruturas de teoria dos grafos. Assim como, as similaridades desse grafo com a classe de grafos cobra triangular alternada.

6.2. Outras Contribuições Envolvendo os Jogos de Torres Explorados

No decorrer deste trabalho, algumas decisões foram necessárias para tornar os testes baseados em torres em jogos combinatórios. Dentre elas, pode-se relatar a definição de um estado final para a Torre de Londres e o Algoritmo 7 para solucionar o jogo, as propriedades dos grafos e o total de vértices deste.

Na Torre de Oxford, foram deliberadas duas versões. Na primeira, o objetivo do jogo é passar todos os n discos de mesmo tamanho e cores diferentes, de p_0 para p_2 , sem se preocupar com a ordem das cores. Para essa versão foi gerado um algoritmo iterativo (Algoritmo 8), com a complexidade computacional $O(n)$. Na segunda versão da ToO, o objetivo do jogo é passar os n discos de mesmo tamanho, de um estado inicial aleatório para o estado final em p_0 , obedecendo a ordem das cores predefinida. Assim, gerou-se um algoritmo recursivo (Algoritmo 9) para solucionar o jogo.

Com isso, nota-se que os jogos combinatórios com 3 torres (ou pilhas) são de fato exponenciais no número de movimentos. Mas, decidir qual o próximo movimento pode ser feito tem tempo linear. Caso o jogador saiba que está no caminho ótimo ou qual foi o último movimento realizado, passa a ser constante. Isso significa que esse tempo exponencial se dá apenas pela grande quantidade de movimentos necessários e não pela dificuldade de tomar uma decisão.

6.3. Análise Comparativa sobre os Grafos das Variações dos Jogos de Torres Explorados

Nesta seção, discute-se as propriedades gerais dos grafos associados aos estados da variação proposta da Torre de Hanoi a Torre de Manaus, comparando com todas as variações abordadas neste trabalho: Torre de Oxford, Torre de Londres e Torre de Bucareste. Na Tabela 6.1, faz-se uma comparação dos aspectos teóricos observados nos grafos das principais variações da Torre de Hanoi, encontradas na literatura, no que diz respeito ao número de vértices, número de arestas, Δ (grau máximo), δ (grau mínimo) e se o grafo é conexo.

Tabela 6.1. Análise dos grafos das Torres exploradas.

Torre		Vértices	Arestas	Δ	δ	Conexo
Discos	Nome					
2	ToH	9	12	3	2	Sim
	ToB	9	8	2	1	Sim
	ToO	12	18	4	2	Sim
	ToL	10	14	4	2	Sim
	ToM	3	1	1	0	Não
3	ToH	27	39	3	2	Sim
	ToB	27	26	2	1	Sim
	ToO	60	108	6	2	Sim
	ToL	36	54	4	2	Sim
	ToM	19	22	3	1	Não

Observa-se que o grafo do jogo proposto, a Torre de Manaus, é o único desconexo, sempre tendo dois componentes conectados. O primeiro componente é formado por vértices bons, que representam estados iniciais: levam ao estado final do jogo, de acordo com a jogabilidade proposta; e a segunda componente é formada por vértices ruins, esta-

dos regulares que não levam ao estado final do jogo. Isso fica muito claro no grafo M_3^3 mostrado na Figura 5.4b.

São dadas as representações, na Figura 6.11, dos grafos do estado da arte investigado, com seu desenho no plano. Pode-se observar que o grafo O_3^3 é simples, não-direcionado, conexo e é composto por subgrafos da classe de grafo cobra-triangular (*triangular snake graph* - T_3 ou grafo borboleta (*butterfly graph*). Pode-se observar, também, que o grafo O_3^3 é o mais geral, sendo um supergrafo de todos os demais. L_3^3 possui um grafo simples, não-direcionado, conexo, classe de grafo cobra-triangular ou borboleta.

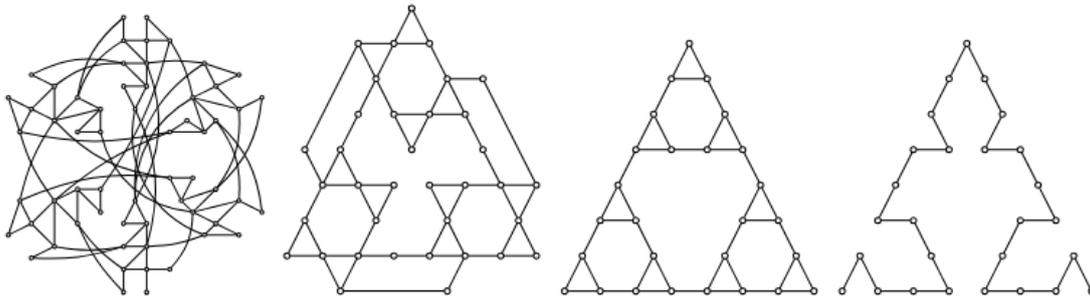


Figura 6.11. Grafos das torres exploradas: O_3^3 , L_3^3 , H_3^3 e B_3^3 .

Analisando o grafo H_3^3 , classificado como simples, não-direcionado, conexo, formado por subgrafos C_3 e é baseado no triângulo de Sierpinski. Finalizando, B_3^3 é um grafo simples, não-direcionado, conexo e é um grafo caminho $P_{estados}$. O grafo B_3^3 , do jogo ToB , este é um subgrafo de H_3^3 , que pode ser visto claramente comparando seus grafos.

No entanto, os grafos H_3^3 e L_3^3 , da ToH e ToL , respectivamente, possuem uma forte interseção, mas um não é subgrafo do outro. A Figura 6.12, traz a comparação dos grafos H_3^3 e L_3^3 para ilustrar esta afirmação. Os vértices em vermelho são os estados da Torre de Londres que não pertencem à Torre de Hanoi. Enquanto que os verdes representam os estados regulares tanto no jogo da Torre de Hanoi quanto na Torre de Londres. Em cinza, os vértices e arestas que pertencem somente ao jogo da Torre de Hanoi.

Finalizando, apresenta-se na Figura 6.13 para uma comparação entre os grafos H_3^3 e L_3^3 com o M_3^3 . Pode-se observar que, mesmo para instâncias pequenas, com discos $n = 3$, M_3^3 é o único desconexo. Os vértices do M_3^3 são particionados em $|V_b|$, vértices bons, que representam os estados iniciais do jogo, e V_r , vértices ruins, que representam os estados proibidos, pois partindo deles não é possível chegar ao estado final do jogo.

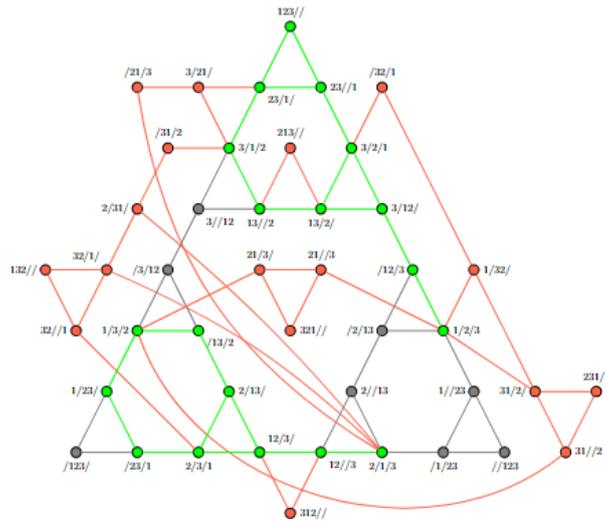


Figura 6.12. Comparação entre os grafos da ToH e ToL: em verde os vértices pertencentes a ambos os grafos, em vermelho os que pertencem apenas ao L_3^3 e em cinza os que pertencem somente ao H_3^3 .

Além disso, M_3^3 diferente de L_3^3 , é um subgrafo de H_3^3 .

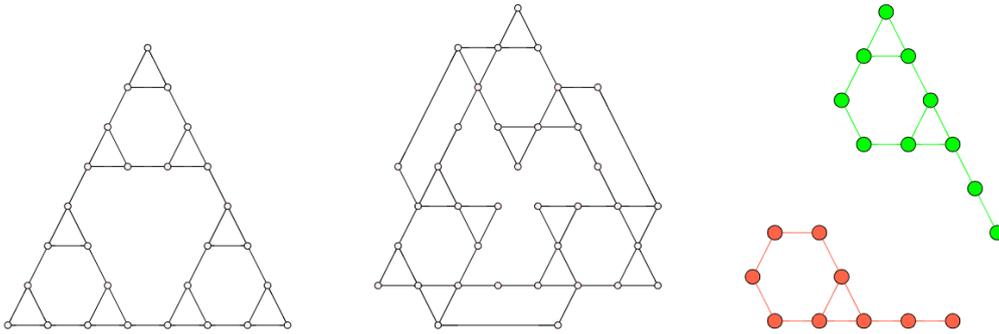


Figura 6.13. Comparação entre os grafos H_3^3 , L_3^3 e M_3^3 .

Capítulo 7

Considerações Finais

Esta dissertação trouxe a proposta do jogo da Torre de Manaus, um jogo que une as regras da Torre de Hanoi e a Torre de Londres. O foco deste trabalho foi jogos combinatórios baseados em torres, tal como o jogo clássico da Torre de Hanoi, tomado como guia, dada a sua popularidade e todo o estado da arte desenvolvido nesse jogo.

Inicialmente, explorou-se a Torre de Hanoi desde as regras do jogo, estados regulares, perfeitos e irregulares, número de movimentos mínimos para a solução ótima do problema, algoritmos recursivo, iterativo, detecção de desvio do caminho ótimo e a complexidade de tempo $O(2^n)$. Além disso, estudou-se a caracterização dos grafos baseados no Triângulo de Sierpinski e, por fim, as aplicações da ToH, que se obteve como fruto a publicação LAWCG [Martins et al. 2022]. A partir do arcabouço adquirido pelo mapeamento da ToH, buscou-se outras variações desse jogo e, quanto à classificação definida, encontrou-se como principais: a Torre de Bucareste, a Torre de Londres e a Torre de Oxford.

Sobre a ToB foram compreendidas as regras do jogo e o número mínimo de movimentos para solucioná-lo. Produziu-se os algoritmos iterativo e recursivo e comprovou-se que são necessários $3^n - 1$ movimentos para solucionar o jogo. A ToL, utilizada em testes neurocognitivos, é diferente da ToH devido à capacidade de seus pinos. Esse jogo não possuía um estado final, esse estado era recebido por meio de um cartão problema. Então, neste trabalho a ToL passou por algumas adaptações, a essencial foi o estado final com todas as esferas em p_0 .

A ToO, se difere da ToH por relaxar as restrições, pois este jogo só possui uma

regra. Na literatura, o jogador sempre sairá de uma posição inicial e deverá chegar a uma configuração recebida em um cartão problema e deverá solucioná-lo com o menor número de movimentos possíveis. Porém, nesta pesquisa, para essa torre ser estudada como um jogo combinatório, criou-se duas versões, ambas guardando a essência do jogo, no caso, somente uma regra, mas agora com um estado final definido. Com isso, também gerou-se o algoritmo de resolução e a modelagem em grafos para esse jogo.

Após o mapeamento das variações da ToH, fez-se uma nova publicação no CLAIO [Martins et al. 2022]. Respondendo à Hipótese de Pesquisa, desta dissertação, foi apresentada uma nova variação da ToH: o jogo da Torre de Manaus(ToM), mas como jogo da Torre de Hanoi-Londres, por ainda estar em processo de desenvolvimento. Nesse jogo foi possível observar que as regras e estrutura combinadas da Torre de Hanoi e da Torre de Londres resultam em uma jogabilidade mais desafiadora. A Torre de Manaus foi apresentada no SBPO [Martins e Freitas 2023] e será apresentada no LAWCG novamente [Martins, Costa e Freitas 2024], porém, ainda como jogo da Torre de Hanoi-Londres. Desse jogo, foram generalizados os estados irregulares, regulares bons e regulares ruins. Além de serem definidas as propriedades do grafo, sendo a principal que o grafo é desconexo. Também foi indicado o total de vértices, os vértices bons, os vértices ruins e foi caracterizada a estrutura do grafo.

Por fim, tem-se a generalização do grafo da Torre de Hanoi por estruturas em grafos [Martins, Costa e Freitas 2024], definindo-se a operação de enxerto e realizando a construção de H_3^n por meio de algoritmo recursivo com esta operação. Os algoritmos das variações da Torre de Hanoi, as similaridades do grafo H_3^n com o cobra triangular alternada e a contribuição no jogo criado inCircle [Freitas et al. 2019], entre outros. Como trabalhos futuros, propõe-se a caracterização dos grafos da ToL e da ToO.

Referências

- ALLARDICE, R.; FRASER, A. La tour d'hanoï. *Proceedings of the Edinburgh Mathematical Society*, Cambridge University Press, v. 2, p. 50–53, 1883.
- ANDRADE, T. et al. Implementação do jogo torre de hanói como estratégia de aprendizagem de pilha. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2016. v. 27, n. 1, p. 1311.
- BENJAMIN, A.; CHARTRAND, G.; ZHANG, P. *The Fascinating World of Graph Theory*. [S.l.]: Princeton University Press, 2017.
- BEREND, D.; SAPIR, A. The cyclic multi-peg tower of hanoi. *ACM Transactions on Algorithms (TALG)*, ACM New York, NY, USA, v. 2, n. 3, p. 297–317, 2006.
- BERG, W. K.; BYRD, D. L. The tower of london spatial problem-solving task: Enhancing clinical and research implementation. *Journal of Clinical and Experimental Neuropsychology*, Taylor & Francis, v. 24, n. 5, p. 586–604, 2002.
- BERLEKAMP, E. R.; CONWAY, J. H.; GUY, R. K. *Winning Ways for Your Mathematical Plays, vol. 1*. [S.l.]: AK Peters Ltd, 2001.
- BRANDSTÄDT, A.; LE, V. B.; SPINRAD, J. P. *Graph classes: a survey*. [S.l.]: SIAM, 1999.
- CORMEN, T. *Desmistificando algoritmos*. [S.l.]: Elsevier Brasil, 2017. v. 1.
- CORMEN, T. H. et al. *Algoritmos Teoria e Prática. Tradução Arlete Simille Marques*. [S.l.]: Rio de Janeiro: Elsevier, 2012.
- DEMAINE, E. D. Playing games with algorithms: Algorithmic combinatorial game theory. In: SPRINGER SCIENCE & BUSINESS MEDIA. *Mathematical Foundations of Computer Science 2001: 26th International Symposium, MFCS 2001 Mariánské Lázně, Czech Republic, August 27-31, 2001 Proceedings*. [S.l.], 2001. v. 2136, p. 18.
- DIESTEL, R. *Graph Theory*. [S.l.]: Springer, 2017.
- EPPSTEIN, D.; FRISHBERG, D.; MAXWELL, W. On the treewidth of hanoi graphs. *Theoretical Computer Science*, Elsevier, v. 906, p. 1–17, 2022.
- ER, M. An analysis of the generalized towers of hanoi problem. *BIT Numerical Mathematics*, Springer, v. 23, p. 429–435, 1983.

- ER, M. Performance evaluations of recursive and iterative algorithms for the towers of hanoi problem. *Computing*, Springer, v. 37, n. 2, p. 93–102, 1986.
- ER, M. An optimal algorithm for reve’s puzzle. *Information sciences*, Elsevier, v. 45, n. 1, p. 39–49, 1988.
- FERGUSON, T. S. Game theory, optimal stopping, probability and statistics: Papers in honor of thomas s. ferguson. In: IMS. [S.l.], 2000.
- FREITAS, R. D. et al. Incircle: uma versão planar do jogo da torre de hanoi na aprendizagem de conceitos matemático-computacionais. In: *XVIII SBGames*. [S.l.: s.n.], 2019. p. 1168–1171.
- GAREY, M. R.; JOHNSON, D. S. *Computers and intractability*. [S.l.]: Publisher Freeman San Francisco, 1979. v. 174.
- HAGE, J. Enumerating submultisets of multisets. *Information processing letters*, Elsevier, v. 85, n. 4, p. 221–226, 2003.
- HEARN, R. A.; DEMAINE, E. D. *Games, puzzles, and computation*. [S.l.]: CRC Press, 2009.
- HEMPEL, L.; SCHMIEDEL, R.; KÄMMERER, L. Operations research in logistics. *Perspectives on Operations Research: Essays in Honor of Klaus Neumann*, Springer, p. 255, 2006.
- HERTER, F.; ROTE, G. Loopless gray code enumeration and the tower of bucharest. *Theoretical Computer Science*, Elsevier, v. 748, p. 40–54, 2018.
- HINZ, A. M.; KLAVŽAR, S.; PETR, C. *The Tower of Hanoi - Myths and Maths*. 2^a. ed. [S.l.]: Springer, 2018.
- HINZ, A. M.; KLAVŽAR, S.; ZEMLJIČ, S. S. A survey and classification of sierpiński-type graphs. *Discrete Applied Mathematics*, Elsevier, v. 217, p. 565–600, 2017.
- HINZ, A. M. et al. A mathematical model and a computer tool for the tower of hanoi and tower of london puzzles. *Information Sciences*, Elsevier, v. 179, n. 17, p. 2934–2947, 2009.
- HOFSTADTER, D. R. Metamagical themas. *Scientific American*, JSTOR, v. 248, n. 5, p. 16–E18, 1983.

- HUANG, S.-H.; LIN, T.-H. Heuristic algorithms for container pre-marshalling problems. *Computers & Industrial Engineering*, Elsevier, v. 62, n. 1, p. 13–20, 2012.
- JEYANTHI, P.; MAHESWARI, A.; VIJAYALAKSHMI, M. 3-product cordial labeling of some snake graphs. *Proyecciones (Antofagasta)*, SciELO Chile, v. 38, n. 1, p. 13–30, 2019.
- KÄMMERER, L.; HEMPEL, L. The towers of hanoi and pileproblems. In: SPRINGER. *Operations Research Proceedings 1996*. [S.l.], 1997. p. 84–89.
- KLAVŽAR, S.; PETERIN, I.; ZEMLJIČ, S. S. Hamming dimension of a graph—the case of sierpiński graphs. *European Journal of Combinatorics*, Elsevier, v. 34, n. 2, p. 460–473, 2013.
- LU, X.-M.; DILLON, T. Parallelism for multipeg towers of hanoi. *Mathematical and computer modelling*, Elsevier, v. 21, n. 3, p. 3–17, 1995.
- MARTINS, L.; COSTA, J.; FREITAS, R. de. Graph aspects and algorithms of a tower of hanoi-london hybrid game. *11th Latin American Workshop on Cliques in Graphs (LAWCG)*, 2024.
- MARTINS, L.; COSTA, J.; FREITAS, R. de. Sobre a generalização do grafo da torre de hanoi. In: SBC. *Encontro de Teoria da Computação (ETC)*. [S.l.], 2024. p. 110–113.
- MARTINS, L.; FREITAS, R. de. Jogos baseados em torres de diferentes capacidade. *Anais do Simpósio Brasileiro de Pesquisa Operacional*, Galoá, v. 55, n. 160951, 2023.
- MARTINS, L. et al. Some variations of the tower of hanoi and their graph properties. *10th Latin American Workshop on Cliques in Graphs (LAWCG)*, 2022.
- MARTINS, L. et al. Tower of hanoi variations, pile problems and o.r. applications. *XXI The Latin-Ibero American Conference on Operations Research (CLAIO)*, 2022.
- MILLA, K. et al. Does movement matter? prefrontal cortex activity during 2d vs. 3d performance of the tower of hanoi puzzle. *Frontiers in Human Neuroscience*, Frontiers Media SA, v. 13, p. 156, 2019.
- MILLER, C. D. "Ch. 4: Binary Numbers and the Standard Gray Code". *Mathematical Ideas*. 9ª. ed. [S.l.: s.n.], 2000.
- MIYAZAWA, F. K. Introdução à teoria dos jogos algorítmica. *UNICAMP, São Paulo*, 2014.

- OLIVEIRA, B.; BRIM, J.; PINHEIRO, N. O jogo torre de hanói como ferramenta mediadora no ensino de potências: um estudo com os alunos do 6o ano do ensino fundamental. *Revista Brasileira de Ensino de Ciência e Tecnologia*, v. 12, n. 01, p. 160–175, 2019.
- OLIVEIRA, S.; CALEJON, L. O jogo torre de hanói para o ensino de conceitos matemáticos. *Rencima. Edição Especial: Educação Matemática*, v. 7, n. 4, p. 149 – 158, 2016. Disponível em: <revistapos.cruzeirodosul.edu.br/index.php/rencima/article/download/1194/846>.
- OTEO, J.; ROS, J. A fractal set from the binary reflected gray code. *Journal of Physics A: Mathematical and General*, IOP Publishing, v. 38, n. 41, p. 8935, 2005.
- PARVILLE, H. de. La tour d’hanoi et la question du tonkin. *La Nature*, v. 12, p. 285–286, 1884.
- PINHO, G. et al. Pensamento computacional no ensino fundamental: Relato de atividade de introdução a algoritmos. In: *Anais do Workshop de Informática na Escola*. [S.l.: s.n.], 2016. v. 22, n. 1, p. 261.
- POOLE, D. G. The towers and triangles of professor claus (or, pascal knows hanoi). *Mathematics Magazine*, Taylor & Francis, v. 67, n. 5, p. 323–344, 1994.
- ROMIK, D. Shortest paths in the tower of hanoi graph and finite automata. *SIAM Journal on Discrete Mathematics*, SIAM, v. 20, n. 3, p. 610–622, 2006.
- SANTIAGO, A. D.; KRONBAUER, A. Um modelo lúdico para o ensino de conceitos de programação de computadores. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2016. v. 27, n. 1, p. 420.
- SCHULZ, S. Algorithmen zur auswahl eines g unstigen palettensortiments {ein Beitrag zur automatengest utzten Palettisierung Diplomarbeit}. *HAB Weimar*, 1989.
- SCORER, R.; GRUNDY, P. M.; SMITH, C. A. B. Some binary games. *The Mathematical Gazette*, JSTOR, v. 28, n. 280, p. 96–103, 1944.
- SEOUD, M.; FARID, M.; ANWAR, M. Some difference graphs. *arXiv preprint arXiv:2209.07317*, 2022.
- SHALLICE, T. Specific impairments of planning. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, The Royal Society London, v. 298,

- n. 1089, p. 199–209, 1982.
- SIERPINSKI, W. Sur une courbe dont tout point est un point de ramification. *CR Acad. Sci.*, v. 160, p. 302–305, 1915.
- SILVA, L. E. C. D.; LIMBERGER, L. S.; PELLANDA, N. M. C. Torre de hanói: possibilidades na construção do conhecimento. *Proceedings of SBGames - SBC*, 2014. ISSN 2179-2259.
- SILVA, R. J. M. et al. Explorando a matemática do jogo torre de hanói. Universidade Federal Rural do Semi-Árido, 2018.
- SOUZA, R. d. O. et al. Contribuição a neuropsicologia do comportamento executivo: torre de londres e teste de wisconsin em indivíduos normais. *Arquivos de Neuro-psiquiatria*, SciELO Brasil, v. 59, n. 3A, p. 526–531, 2001.
- SUCH, S. C. et al. *Comparação entre as versões manuais e computadorizadas dos testes Torre de Londres (ToL), Teste de STROOP (STROOP) e Teste Wisconsin de Classificação de Cartas (WCST)*. 2011. Monografia (Especialização em Neurociência e suas Fronteiras), UFMG (Universidade Federal de Minas Gerais), Minas Gerais, Brazil.
- SZWARCFITER, J. L. *Teoria Computacional de Grafos: Os Algoritmos*. [S.l.]: Elsevier Brasil, 2018.
- UNTERRAINER, J. et al. What is in a name: Comparing the tower of london with one of its variants. *Cognitive Brain Research*, Elsevier, v. 23, n. 2-3, p. 418–428, 2005.
- WARREN, H. S. "Section 5-4: Counting Trailing 0's". *Hacker's Delight*. 1^a. ed. [S.l.]: Boston MA: Addison-Wesley, 2003.
- YANG, T. et al. Fbbm: a new backup method with data de-duplication capability. *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, IEEE, p. 30–35, 2008.
- ZHANG, Z. et al. The number and degree distribution of spanning trees in the tower of hanoi graph. *Theoretical Computer Science*, Elsevier, v. 609, p. 443–455, 2016.

Apêndice A

Representação Binária

O sistema numérico binário, ou seja os números na base 2, permite-nos fazer uma análise bastante aprofundada do jogo Torre de Hanoi, já que os movimentos dos discos, assim como o da Torre de Hanoi, podem ser facilmente convertidos para esse sistema de numeração em questão e que, por sinal, são números bem particulares [Silva et al. 2018].

As posições dos discos podem ser determinadas mais diretamente a partir do número binário do movimento, o estado inicial sendo o movimento 0, com todos os dígitos 0 e o estado final com todos os dígitos 1, usando as seguintes regras [Warren 2003]:

- há um dígito binário (bit) para cada disco;
- o bit mais significativo (mais à esquerda) representa o maior disco.
- a cadeia de bits é lida da direita para a esquerda;
- um bit, com o mesmo valor que o anterior, significa que o disco correspondente está empilhado em cima do disco anterior na mesmo pino, ou seja, uma sequência direta de 1s ou 0s significa que os discos correspondentes estão todos no mesmo pino;
- um bit com valor diferente do anterior significa que o disco correspondente está uma posição à esquerda ou à direita do anterior.

Para entender melhor como funciona a representação binária, deve-se observar a Tabela A.1, pois, cada posição do número binário está ligada a um disco da Torre de Hanoi e lê-se da direita para a esquerda, procurando o número 1, que atua como um ativador do movimento, onde ele estiver indica que desse disco deve ser movimentado.

Tabela A.1. Contador Binário, onde o número da jogada em binário indica o movimento que deve ser realizado.

Jogadas	Movimentos				Jogadas	Movimentos			
	Disco	Disco	Disco	Disco		Disco	Disco	Disco	Disco
	4	3	2	1		4	3	2	1
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1
8	1	0	0	0					

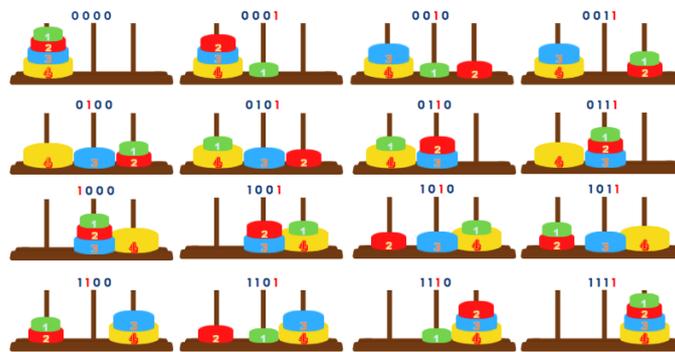


Figura A.1. Torre de Hanoi com solução binária.

Na Figura [A.1](#), explica a resolução do jogo com 4 discos, de acordo com a representação binária do número da jogada relacionado com o que já foi representado na Tabela [A.1](#). Pode-se comprovar que é possível solucionar o jogo com o número mínimo de movimentos se guiando somente com o valor da jogada em binário.

Apêndice B

Código Gray

O sistema numérico binário do código Gray, exposto na Tabela [B.1](#), oferece uma maneira alternativa de resolver o quebra-cabeça [\[Oteo e Ros 2005\]](#). No código Gray, os números são expressos em uma combinação binária de 0s e 1s, mas em vez de ser um sistema numérico posicional padrão, o código Gray opera na premissa de que cada valor difere de seu predecessor por apenas um (e exatamente um) bit alterado [\[Hage 2003\]](#).

Tabela B.1. Construção do Código Gray.

Decimal	Binário	Gray	Decimal	Binário	Gray
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Em código Gray um tamanho de bit é igual ao número de discos em uma determinada Torre de Hanoi, que começa em zero e conta para cima, como explorado na Figura [B.1](#). Assim, o bit alterado a cada movimento corresponde ao disco a ser movido, sendo que, lendo da esquerda para a direita, o último dígito é o bit menos significativo e representa o menor disco, e o primeiro dígito é bit mais significativo e representa o maior disco [\[Miller 2000\]](#).

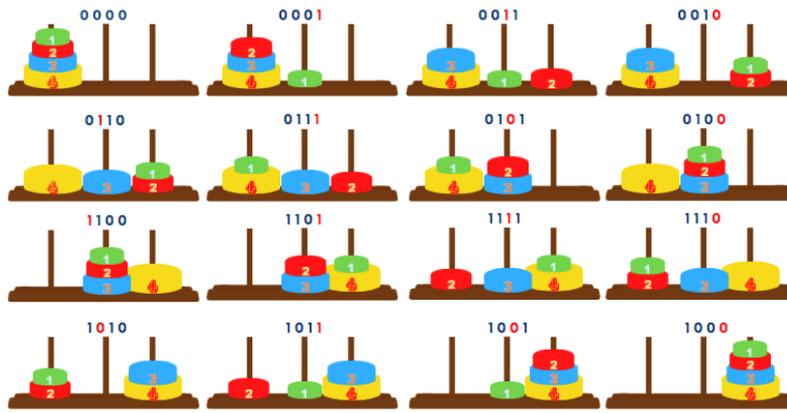


Figura B.1. Resolução da Torre de Hanoi com Código Gray.

Essa técnica identifica qual disco deve ser movido, mas não para onde movê-lo. Para o menor disco, há sempre duas possibilidades. Para os outros discos há sempre uma possibilidade, exceto quando todos os discos estão no mesmo pino, porém, nesse caso, ou é o disco menor que deve ser movido ou o objetivo já foi alcançado.

Apêndice C

Triângulo de Sierpinski

Para a construção do Triângulo de Sierpinski, como apresentado na Figura C.1, parte-se de uma área triangular formada por um triângulo equilátero. Em seguida, uni-se os pontos médios de cada lado do triângulo, formando quatro outras áreas triangulares equiláteras em que três delas têm um lado em comum com a área triangular do centro. Depois, retira-se a área triangular central. O processo consiste em repetir indefinidamente o procedimento anterior em relação a cada uma das áreas triangulares obtidas. O Triângulo de Sierpinski é a figura resultante desse processo *infinito* de retiradas.

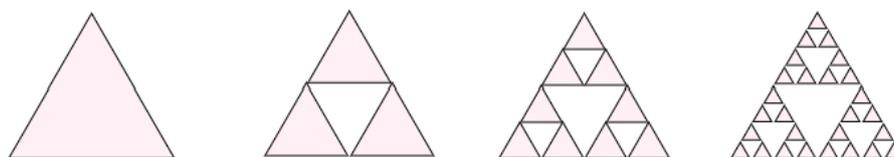


Figura C.1. Primeiros níveis de construção do Triângulo de Sierpinski.

Os grafos do Triângulo de Sierpinski são uma família de grafos que possuem imersão planar cujo limite de $n \rightarrow \infty$. O grafo do Triângulo de Sierpinski é um fractal quando $p = 3$. Para o caso $p = 3$, sabe-se que os grafos de Hanoi $\{H_3^n\}$ estão intimamente relacionados com os grafos do Triângulo de Sierpinski $\{S_3^n\}$ [Klavžar, Peterin e Zemljič 2013]. A imersão planar permite ver a semelhança geométrica entre os grafos do Triângulo de Sierpinski e o grafo da Torre de Hanoi de três pinos [Hinz, Klavžar e Zemljič 2017]. O primeiro grafo do Triângulo de Sierpinski, S_3^3 , é isomórfico a K_3 com uma imersão planar de um triângulo equilátero com lados de comprimento unitário. Os vértices do triângulo coincidem com os vértices de k_3 .

Assumindo, indutivamente, que S_3^{n-1} possui uma imersão planar cuja face externa é embutida geometricamente como um triângulo equilátero, rotula-se os vértices na face externa do triângulo como: v_l, v_r, v_t que são os vértices esquerdo, direito e superior, respectivamente. Para construir S_3^n a partir de S_3^{n-1} , pega-se três cópias de S_3^{n-1} rotuladas $S_3^{n-1,l}, S_3^{n-1,r}, S_3^{n-1,t}$ para os triângulos esquerdo, direito e superior, como apresentado na Figura C.2a e C.2b, e faz-se as seguintes identificações de vértice [Eppstein, Frishberg e Maxwell 2022].

1. Identifica-se v_l em $S_3^{n-1,r}$ com v_r em $S_3^{n-1,l}$, e rotula-se o vértice resultante de v_{lr} .
2. Identifica-se v_t em $S_3^{n-1,l}$ com v_l em $S_3^{n-1,t}$, e rotula-se o vértice resultante de v_{lt} .
3. Identifica-se v_t em $S_3^{n-1,r}$ com v_r em $S_3^{n-1,t}$, e rotula-se o vértice resultante de v_{rt} .

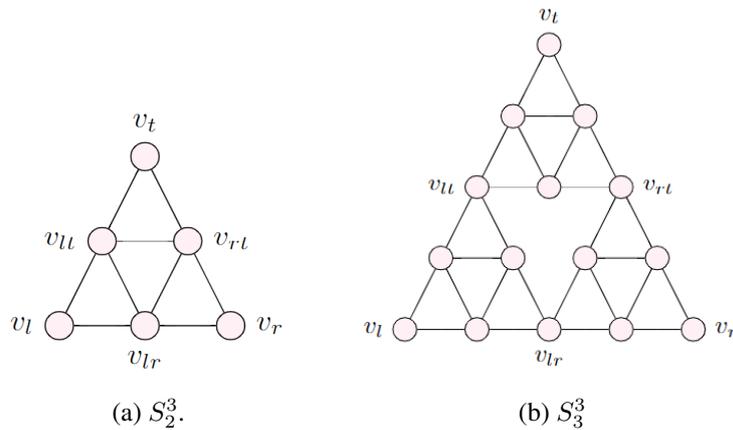


Figura C.2. Triângulo de Sierpinski. Fonte: [Eppstein, Frishberg e Maxwell 2022].

O grafo resultante tem uma imersão plana cuja face externa pode ser imersa novamente como um triângulo equilátero subdividido. Em S_3^n , os vértices esquerdo, direito e superior da face externa estão rotulados em $S_3^{n-1,l}, S_3^{n-1,r}, S_3^{n-1,t}$, respectivamente.

Apêndice D

Problemas de Pilha na Gestão de Logística

Imagine um porto onde há vários contêineres e é necessário carregar ou descarregar cargas dos navios porta-contêineres, Figura [D.2](#). Suponha grandes peças de concreto para uma ponte, Figura [D.1a](#). Ou ainda a construção de um edifício moderno, Figura [D.1b](#), em que a arquitetura foi projetada de forma diferente e manter o equilíbrio das vigas e colunas é uma tarefa difícil. Por fim, pense num desvio de trens em um pátio de triagem.



(a) Construção de uma ponte.



(b) Construção de um edifício.

Figura D.1. Exemplos de pilhas na Gestão de Logística.

Um problema é a organização de contêineres em um navio, Figura [D.2](#), cujo objetivo é encontrar um conjunto de movimentos de contêineres para alcançar um arranjo final que satisfaça certas restrições [[Huang e Lin 2012](#)]. Para solucionar tal problema, Lu e Dillon [1995] propuseram a implementação de paralelismo da Torre de Hanoi, no qual é permitido movimentos simultâneos de discos em uma única etapa. Esses autores analisaram três versões de paralelismo, considerando dois fatores: a ordem de movimentação

dos discos e a utilização de recursos dos pinos. Eles apresentam uma técnica comum para obter a implementação não recursiva, considerando pequenas etapas [Lu e Dillon 1995]. Todos esses problemas podem ser modelados aplicando os princípios básicos da Torre de Hanoi.



Figura D.2. Organização de contêineres em um navio.

Apêndice E

Problemas de *Backup* na Área de TI

Suponha que alguém, que trabalha na área de TI, precisa fazer *backups*, mas tem um número limitado de recursos. Para tal, um esquema de rotação de *backup* de dados em mídias de computador (como fitas) que minimizariam, por reutilização, o número de mídias usadas. Esse é outro exemplo de aplicação da Torre de Hanoi, apresentado na Figura E.1. O esquema determina como e quando cada parte do armazenamento removível é usada para um trabalho de *backup* e por quanto tempo ela é retido depois de ter os dados armazenados nele.

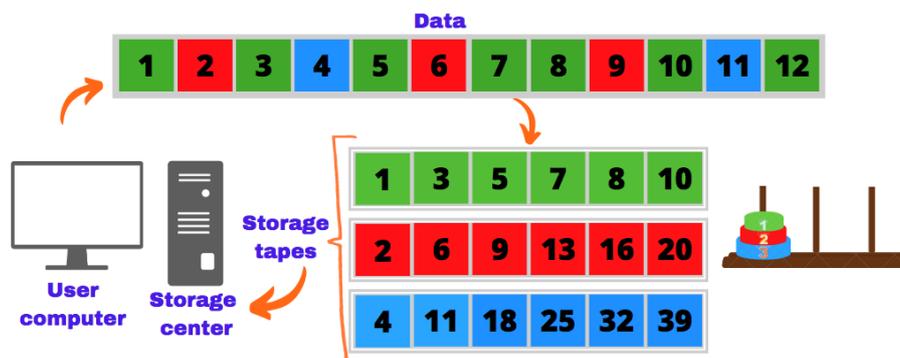


Figura E.1. Exemplo do *backup* em fitas.

Usando um método recursivo para otimizar o ciclo de *backup*, baseando-se na matemática da Torre de Hanoi, cada fita corresponde a um disco no quebra-cabeça e cada movimento do disco para um pino diferente corresponde a um *backup* para essa fita. Assim, em um ciclo de uma semana, a primeira fita pode ser usada, mais frequentemente, a cada 2 dias (1, 3, 5, 7, ...), a segunda poderá ser usada a cada 4 dias (2, 6, 9, ...) e a terceira, a cada 7 dias (4, 11, 18, ...) [Yang et al. 2008].