INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MATO GROSSO

Inteligência Artificial
Lista II - Solução por Meio de Buscas

Vitor Bruno de Oliveira Barth

30 de setembro de 2018

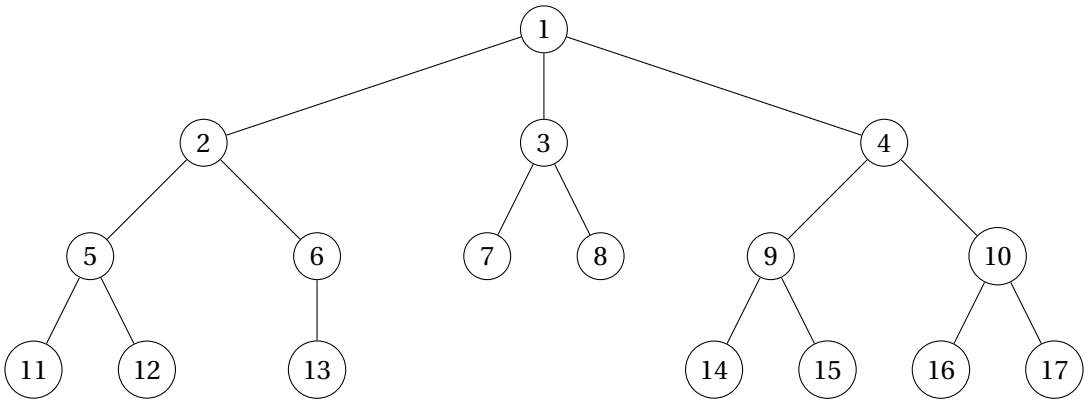
1 NOMEIE OS ALGORITMOS DE BUSCAS QUE RESULTAM DE:

(a) **Busca de Feixe Local, com k = 1**
Hill Climbing

(b) **Busca de Feixe Local, com um estado inicial e sem limite de estados retidos**
Breadth-First Search, mas de forma que cada camada seria expandida em uma única iteração

(c) **Busca de Têmpera Simulada, com temperatura T = ∞ todo o tempo**
Random Search

2 CONSIDERANDO A ÁRVORE ABAIXO, PEDE-SE A ORDEM COM QUE OS NÓS SÃO VISITADOS USANDO-SE OS SEGUINTES ALGORITMOS DE BUSCA:

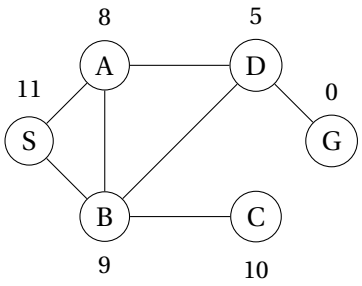


(a) **Breadth-First Search**
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

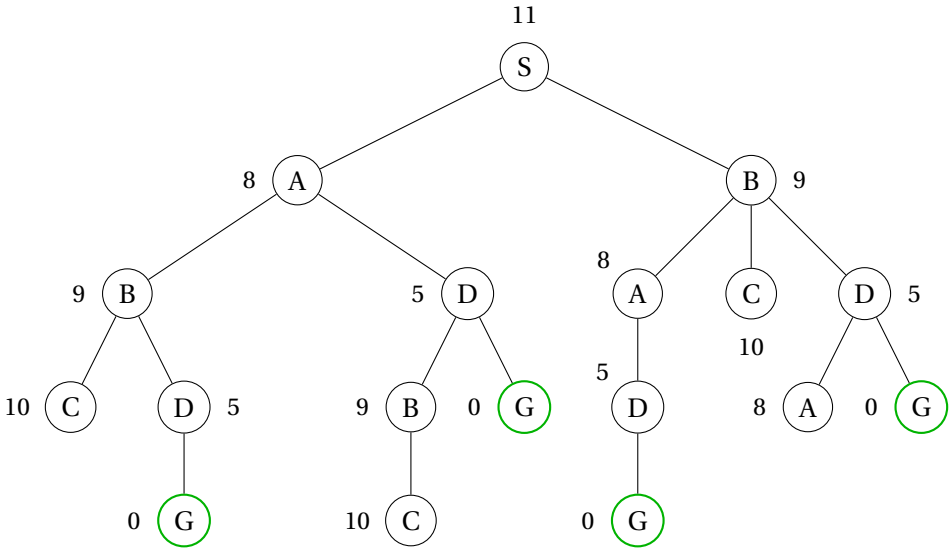
(b) **Depth-First Search**
1, 2, 5, 11, 12, 6, 13, 3, 7, 8, 4, 9, 14, 15, 10, 16, 17

(c) **Depth-First Iterative-Deepening Search**
Caso a profundidade-limite seja maior que três, a execução será idêntica à Depht-First Search

3 DESENHE A ÁRVORE DE BUSCA COMPLETA (DO ESTADO *S* AO *G*) PARA O GRAFO ABAIXO. OS NÚMEROS AO LADO DOS NÓS REPRESENTAM AS DISTÂNCIAS ESTIMADAS DO ESTADO INICIAL (*S*) PARA O ESTADO FINAL (*G*). MOSTRE COMO O PROCEDIMENTO DE BUSCAS PROCEDE NA ÁRVORE QUANDO USANDO:



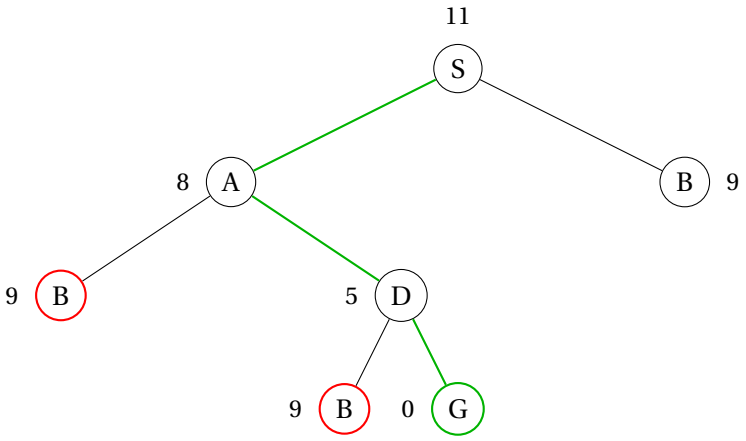
3.1 ÁRVORE DE BUSCA



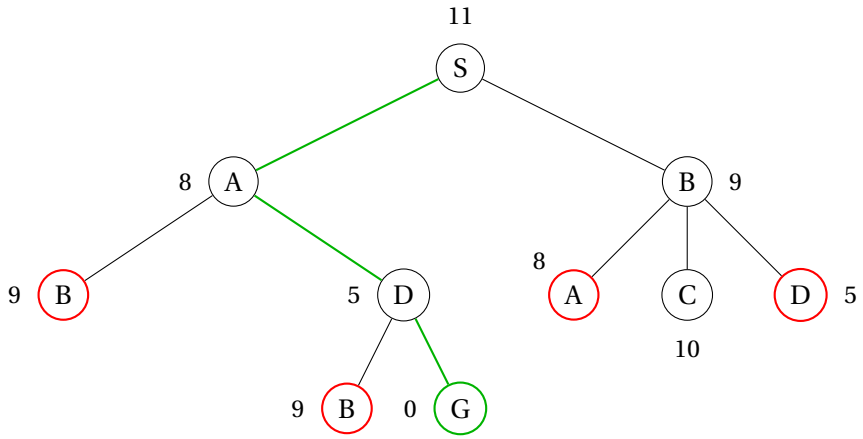
3.2 EXECUÇÃO DOS ALGORITMOS DE BUSCA

Vermelho → Nós já explorados;
Verde → Solução.

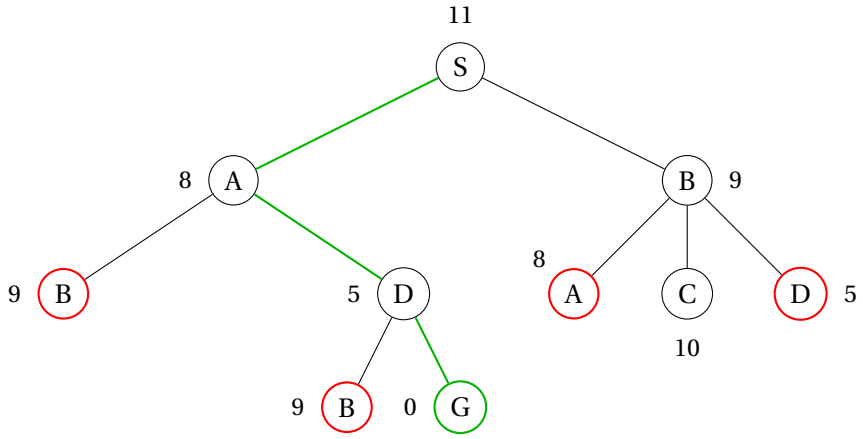
(a) **DFS - Busca em Profundidade**



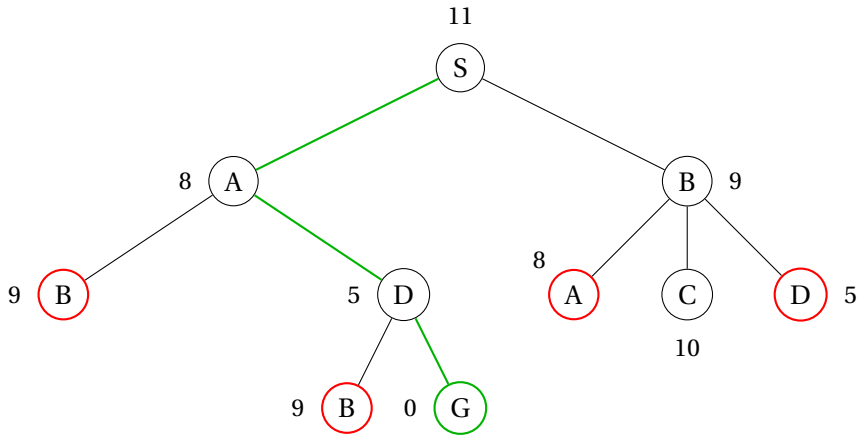
(b) **BFS - Busca em Largura**



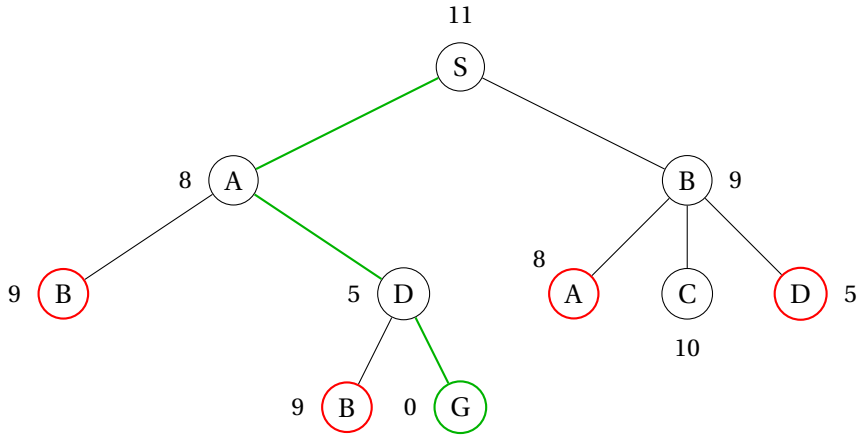
(c) **HC - Subida de Encosta**



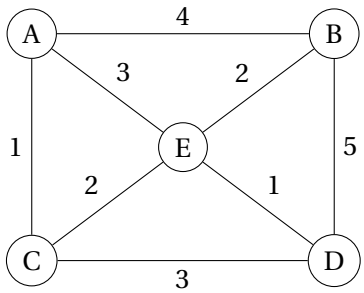
(d) **BS - Feixe-Local com** $k = 2$



(e) **GBFS - Busca Gulosa**



4 Considere o roteiro das cidades A, B, C, D e E:



Cada cidade está conectada a outra cidade por meio de uma estrada, e o tempo de viagem entre as cidades está indicado pelo número mostrado em cada caminho. Suponha que você está na cidade A e quer planejar uma viagem passando por cada cidade uma única vez. Por exemplo, a viagem pelo caminho ABDCEA é uma possível solução que demandaria 17 horas para ser realizada. A sua tarefa é escolher um caminho que minimize o tempo de viagem. Desta forma, pede-se:

(a) **Formule o Espaço de Estados para o problema**

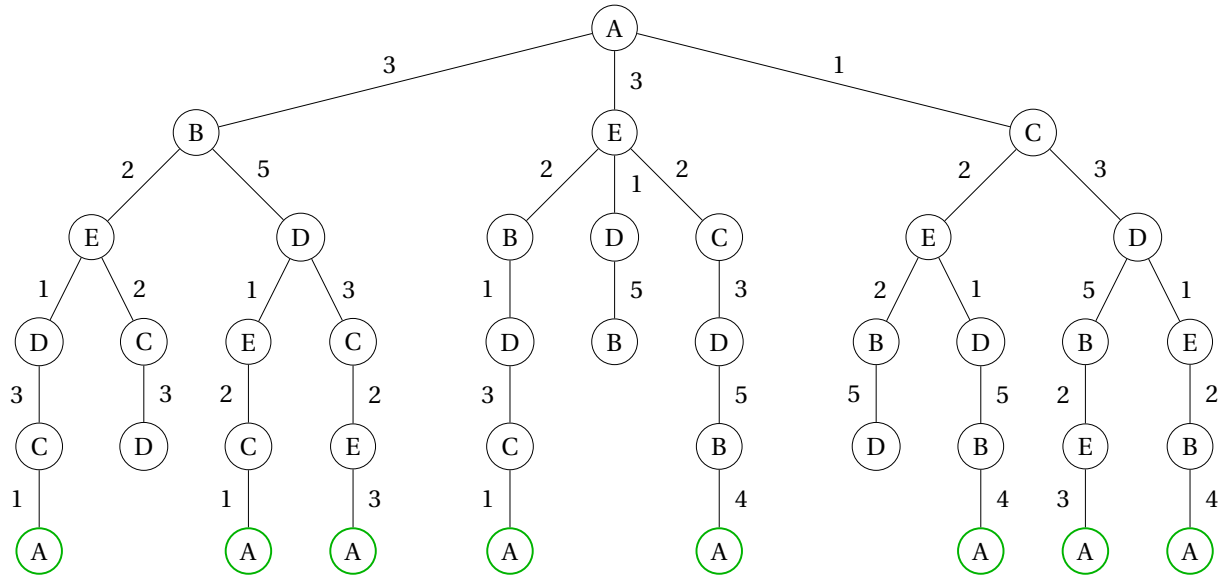
$S: \{A, B, C, D, E\}$

$A: \{A, B, C, D, E\}$

$$Action(s): \begin{cases} s = A \rightarrow \{B, E, C\} \\ s = B \rightarrow \{A, E, D\} \\ s = C \rightarrow \{A, E, D\} \\ s = D \rightarrow \{B, E, C\} \\ s = E \rightarrow \{A, B, D, C\} \end{cases}$$

$$Cost(s,a): cost\_matrix[s,a] \;||\; cost\_matrix = \begin{bmatrix} 0 & 4 & 1 & -1 & 3 \\ 4 & 0 & -1 & 5 & 2 \\ 1 & -1 & 0 & 3 & 2 \\ -1 & 5 & 3 & 0 & 1 \\ 3 & 2 & 2 & 1 & 0 \end{bmatrix}$$

(b) **Esboce o diagrama do Espaço de Estados completo, descrevendo as ações**



(c) **Descreva um algoritmo BFS, e encontre a viagem mais curta da cidade A para a cidade A que visite todas as cidades**

A execução do algoritmo BFS explora a árvore completa, como descrita na questão (b).

A viagem mais curta encontrada é {A, B, E, D, C, A}, que levaria 10 horas para se percorrida.

(d) **Compare os requerimentos de tempo e espaço dos algoritmos de busca-cega DFS, BFS e UCS para este problema**

Fator de Ramificação da Árvore $b = 2$,
Profundidade da Árvore $d = 5$,
Nº de Nós do Grafo $V = 5$,
Nº de Arestas do Grafo $E = 8$.

Tabela 1: Comparação entre os Algoritmos de Busca-Cega.

Algoritmo	Complexidade de Processamento	Complexidade de Espaço
DFS	$O(b^d) \Rightarrow O(32)$	$O(bd) \Rightarrow O(10)$
BFS	$O(b^d) \Rightarrow O(32)$	$O(b^d) \Rightarrow O(32)$
UCS	$O(\|E\| + \|V\| log \|V\|) \Rightarrow O(11.5)$	$O(\|V\|^2) \Rightarrow O(25)$

(e) **Compare os requerimentos de tempo e espaço dos algorimtos heurísticos A*, HC (Hill Climbing/Subida de Encosta) e SA (Simulated Annealing/Têmpera Simulada) para este problema**

Fator de Ramificação da Árvore $b = 2$,
Profundidade da Árvore $d = 5$.

Tabela 2: Comparação entre os Algoritmos de Busca Heurística.

Algoritmo	Complexidade de Processamento	Complexidade de Espaço
A*	$O(b^d) \Rightarrow O(32)$	$O(b^d) \Rightarrow O(32)$
HC	$O(\infty)$	$O(b) \Rightarrow O(2)$
SA	$O((b^2 + b) log b) \Rightarrow O(1.8)$	$O(b) \Rightarrow O(2)$

5 CONSIDERE O PROBLEMA DO QUEBRA-CABEÇA DE 3 PEÇAS, QUE É UMA SIMPLIFICAÇÃO DO QUEBRA-CABEÇA DE 8 PEÇAS. NESSA SIMPLIFICAÇÃO, HÁ 3 PEÇAS E UM ESPAÇO VAZIO, CONFORME MOSTRADO ABAIXO.

Estado Inicial

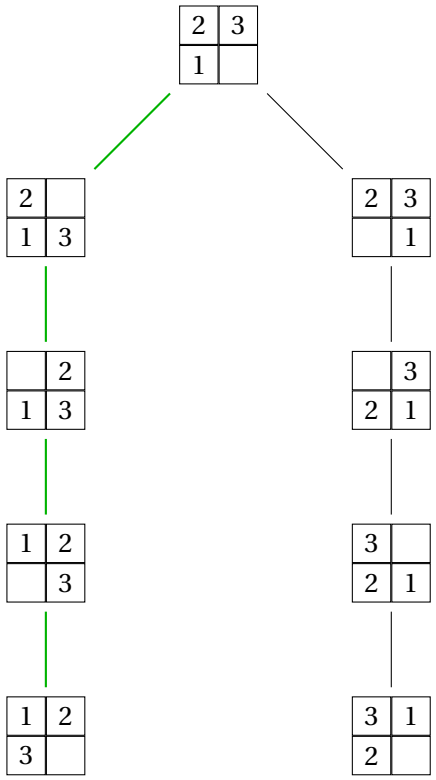
2	3
1	

Objetivo

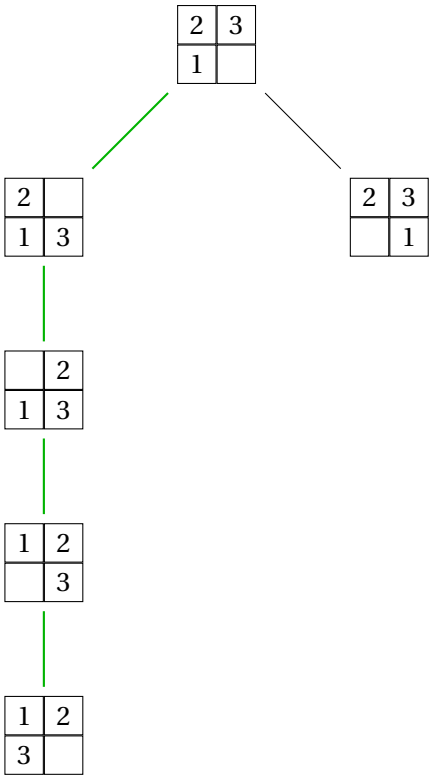
1	2
3	

Há quatro operações possíveis para a peça branca/vazia: acima, abaixo, esquerda ou direita. Dados os Estados Inicial e Objetivo, mostre como o caminho para o objetivo pode ser encontrado usando:

(a) **BFS - Busca em Largura**

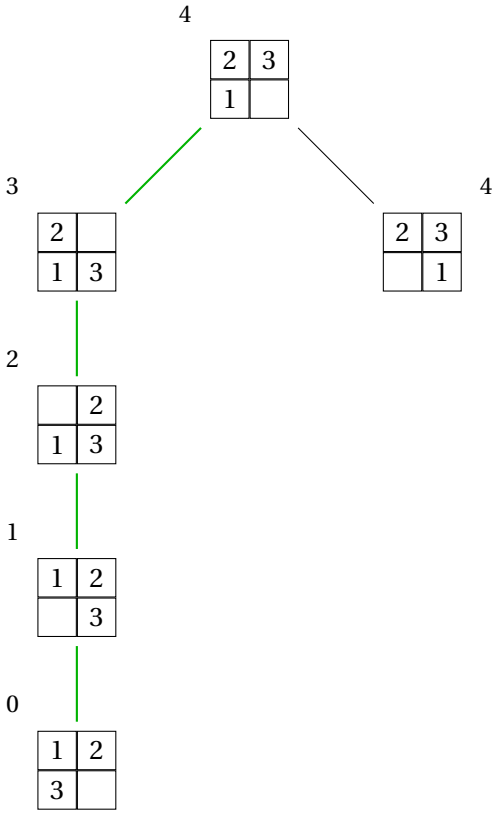


(b) **DFS - Busca em Profundidade**



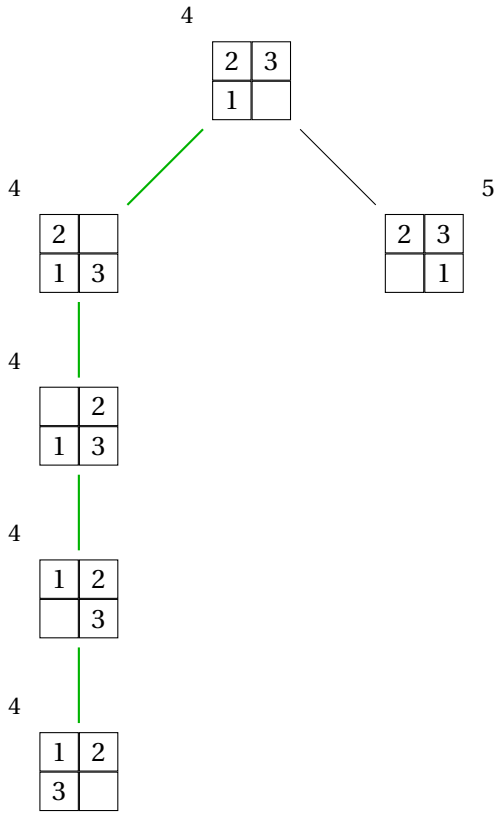
(c) **GBFS - Busca Gulosa**

Usando a Heurística de Manhattan



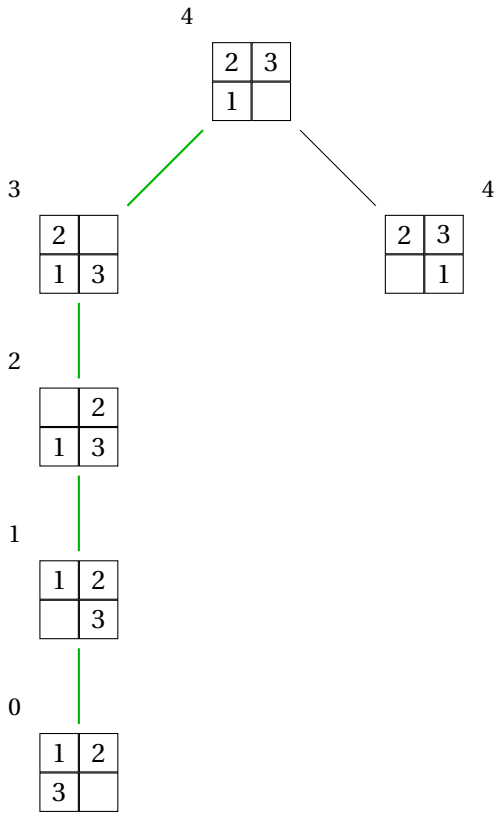
(d) **A***

O valor do nó corresponde à $h(s) + c(s)$, onde $h(s)$ é a Heurística de Manhattan e $c(s)$ é o custo.



(e) **HC - Subida de Encosta**

Usando a Heurística de Manhattan



6 Tendo-se o Estado Inicial abaixo para o Problema das 4 Rainhas, considere a Busca de Subida de Encosta (HC), usando como heurística o número de conflitos diretos ou indiretos entre as rainhas, para encontrar um Estado Válido (sem conflitos).

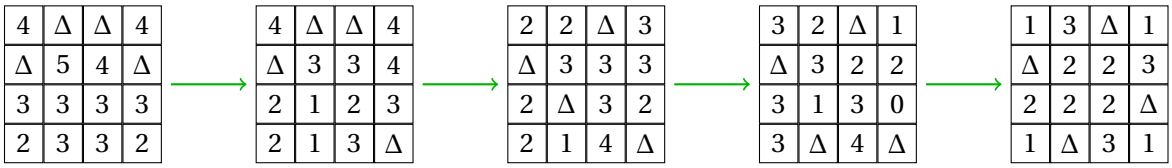
Estado Inicial

4	Δ	Δ	4
Δ	5	4	Δ
3	3	3	3
2	3	3	2

Note que o valor da heurística de um Estado é o número de pares distintos de rainhas que podem se atacar mutuamente. Os sucessores de um estados são todos os estados possíveis gerados pelo movimento de uma única rainha par aoutra posição na mesma coluna. A busca HC escolhe, em cada passo, o sucessor com menor número de conflitos.

(a) **Esboce a Árvore de Estados que Busca HC, usando a heurística de mínimo conflito, expolora deste Estado Inicial. Os sucressores de cada Estado são aqueles estados que têm menos conflitos.**

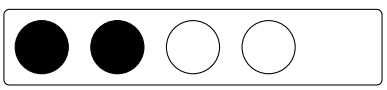
Em caso de haver várias posições com o mesmo número de conflitos, será escolhida a que está mais acima e mais à direita.



(b) **Todos os nós folhas da árvore do item (a) foram gerados nas soluções encontradas para o Problema das 4 Rainhas? Se não, como fazer para que todas as folhas da árvore representem uma solução?**

Não, a Busca por Subida de Encosta não avalia todos os nós folhas. Para que todas as folhas representem soluções, é necessário que a árvore seja completamente explorada, e nenhum dos algoritmos estudados garante explorar completamente a árvore.

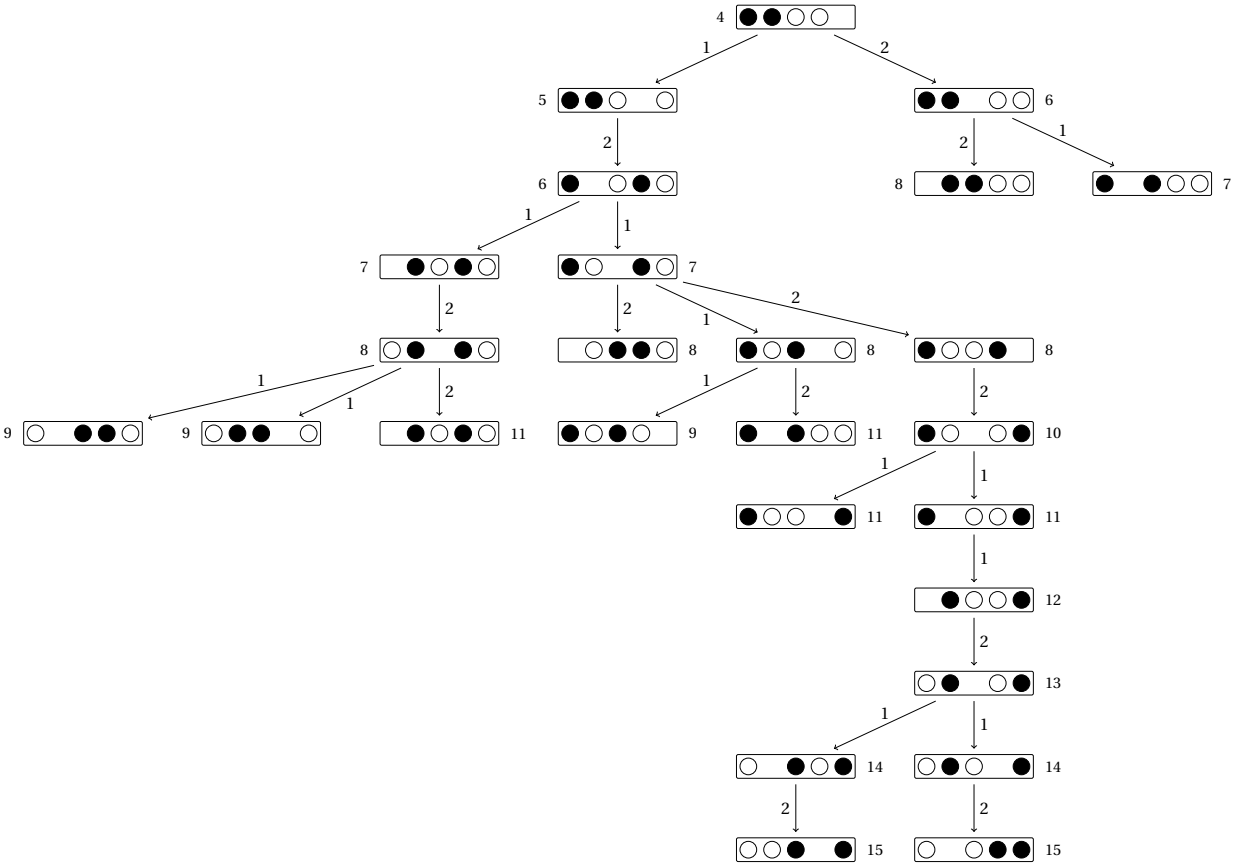
7 Considere o Quebra-Cabeça formado por 5 células, conforme mostra a figura abaixo. As duas primeiras células contêm peças na cor preta, as duas próximas na cor branca, e a última célula está vazia.



Neste Quebra-Cabeça, uma peça pode ser movida para uma célula vazia (custo de uma unidade) ou a peça pode pular sobre no máximo duas células para uma peça vazia (custo igual ao número de células puladas). O objetivo do Quebra-Cabeça é colocar as duas peças pretas à direita das peças brancas, e a peça vazia pode ficar em qualquer posição. Pede-se:

(a) **Resolva o Quebra-Cabeça com Algoritmo A*, adotando a seguinte heurística: $h(s)$ é a soma do número de peças brancas à direita da primeira peça preta e o número de peças brancas à direita da segunda peça branca. Por exemplo, o Estado Inicial da Figura acima possui $h(s) = 4$**

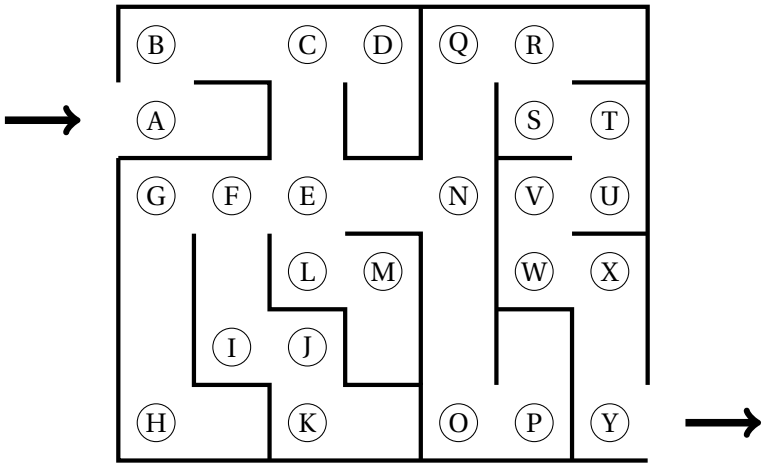
O valor do nó corresponde à $h(s) + c(s)$, onde $h(s)$ é o valor da heurística do nó e $c(s)$ é o custo.



(b) **Mostre que $h(s)$ é admissível, ou seja, $h(s) \leq h*(s)$**

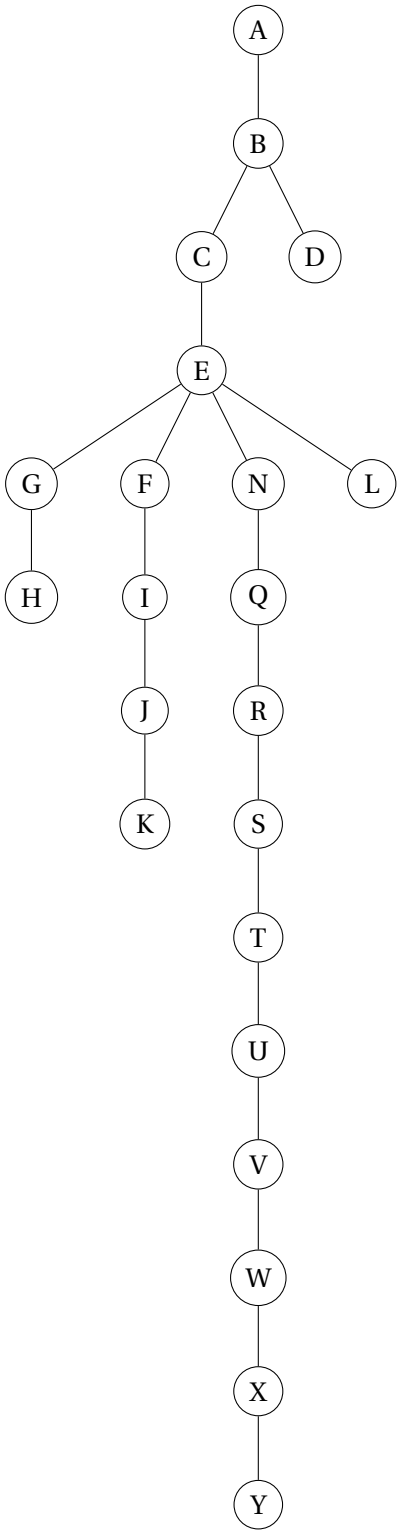
A heurística nunca é maior que o custo do caminho. No primeiro passo, que é o mais distante, a heurística é 4. O custo do Estado Inicial ao Objetivo é 14.

8 A BUSCA DE SOLUÇÃO DE UM CAMINHO DE UM PONTO A OUTRO EM UM LABIRINTO, COMO MOSTRADO NA FIGURA ABAIXO, PODE SER REPRESENTADO POR UMA ÁRVORE DE BUSCA.

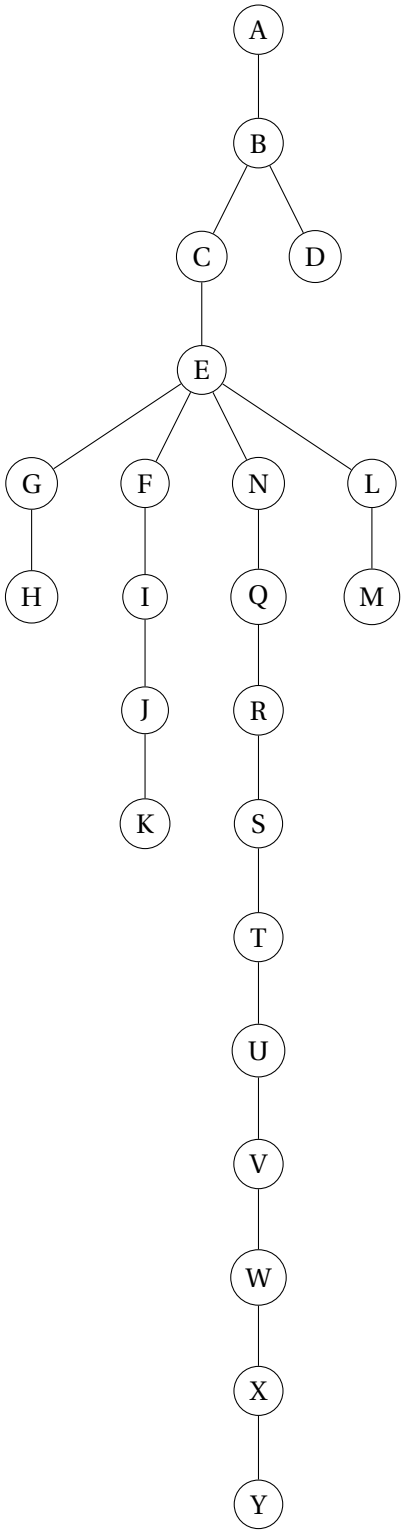


Então, usando os algoritmos de busca DFS, BFS e A*, um de cada vez, encontre o caminho entre a entrada e a saída no labirinto da figura abaixo. Adote uma heurística para o algoritmo A*. Compare o resultado obtido por cada algoritmo em termos de número de estados visitados e tempo de execução.

(a) **DFS**



(b) **BFS**



(c) **A***

Heurística: Distância direita até Y, ignorando barreiras.