

Inteligência Artificial

Lista II - Solução por Meio de Buscas

Vitor Bruno de Oliveira Barth

27 de setembro de 2018

1 NOMEIE OS ALGORITMOS DE BUSCAS QUE RESULTAM DE:

- (a) **Busca de Feixe Local, com $k = 1$**

Hill Climbing

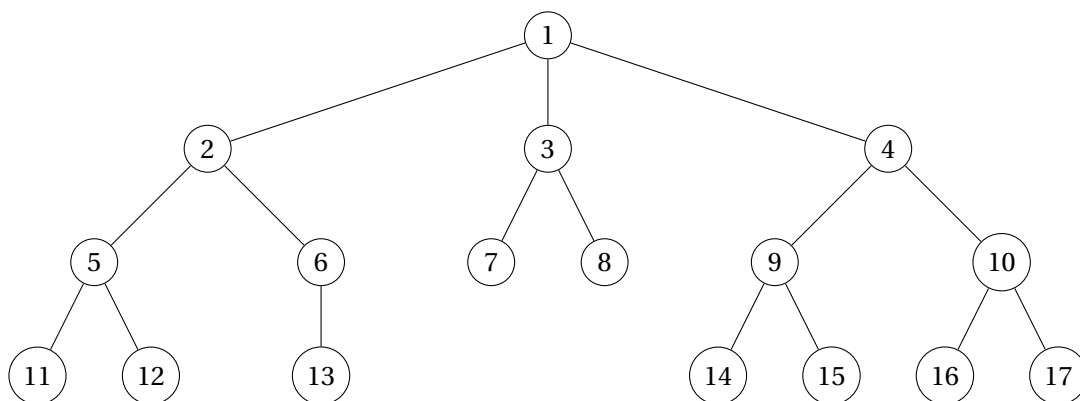
- (b) **Busca de Feixe Local, com um estado inicial e sem limite de estados retidos**

Breadth-First Search, mas de forma que cada camada seria expandida em uma única iteração

- (c) **Busca de Têmpera Simulada, com temperatura $T = \infty$ todo o tempo**

Random Search

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



- (a) **Breadth-First Search**

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

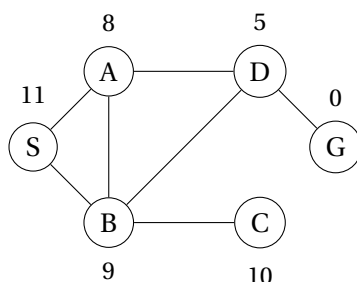
- (b) **Depth-First Search**

1, 2, 5, 11, 12, 6, 13, 3, 7, 8, 4, 9, 14, 15, 10, 16, 17

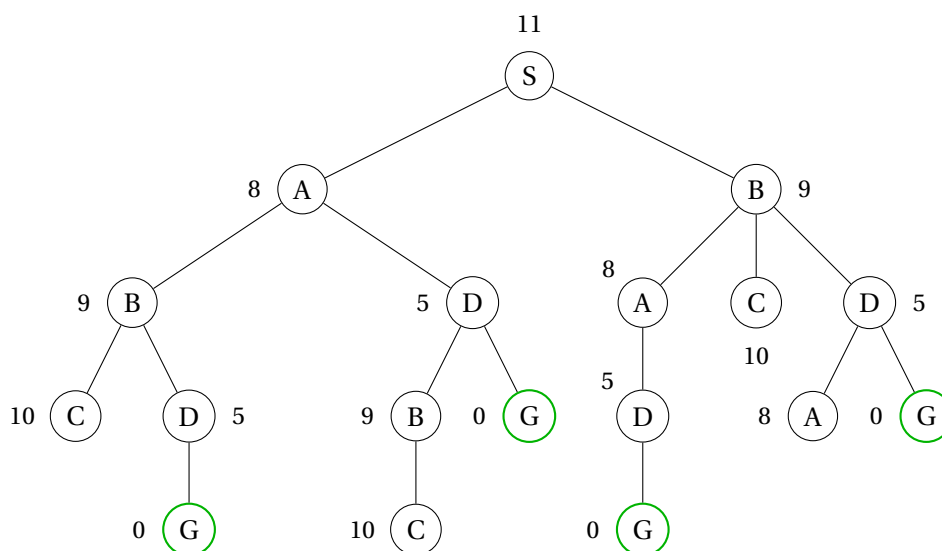
- (c) **Depth-First Iterative-Deepening Search**

Caso a profundidade-limite seja maior que três, a execução será idêntica à Depth-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

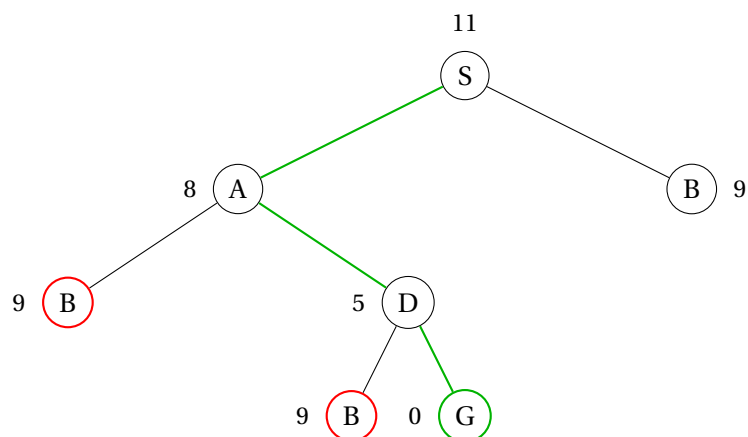


3.2 EXECUÇÃO DOS ALGORITMOS DE BUSCA

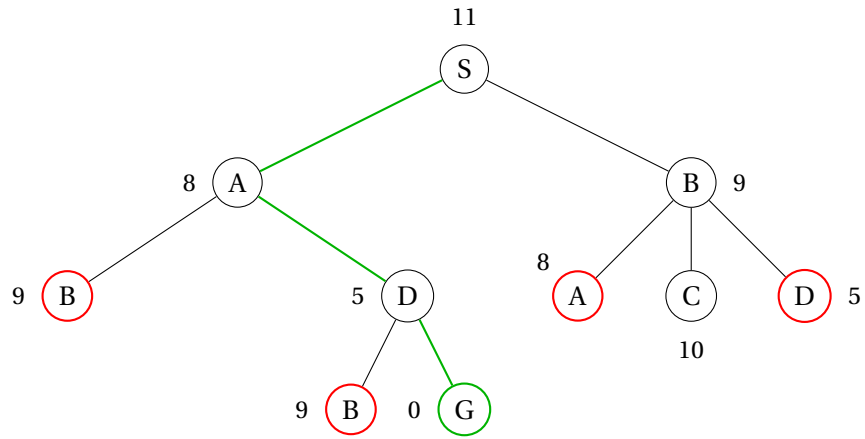
Vermelho → Nós já explorados;

Verde → Solução.

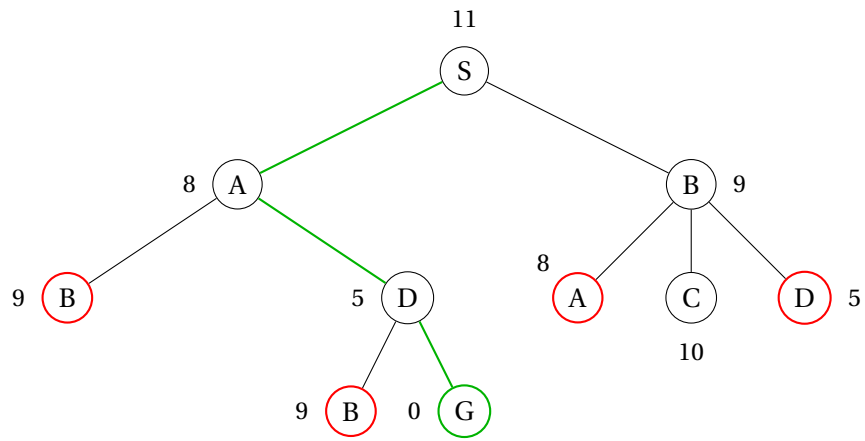
(a) **DFS - Busca em Profundidade**



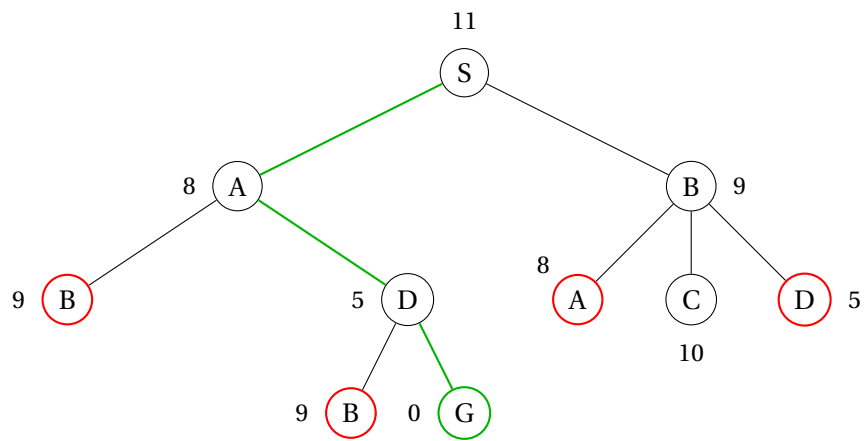
(b) **BFS - Busca em Largura**



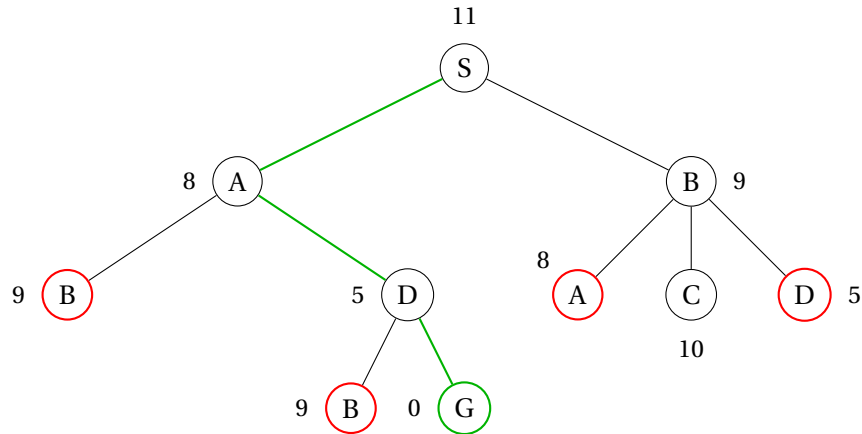
(c) **HC - Subida de Encosta**



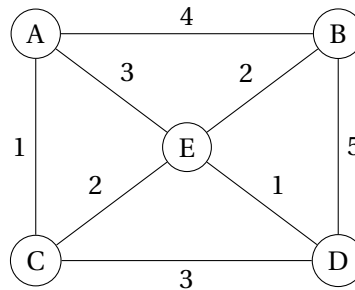
(d) **BS - Feixe-Local com $k = 2$**



(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**

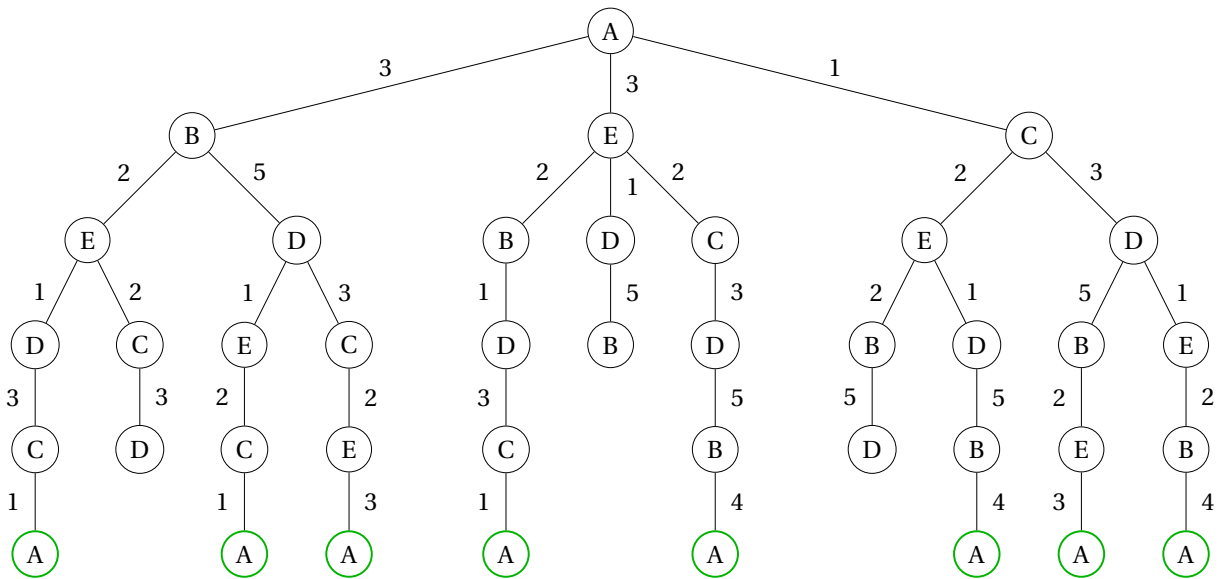
$S: \{A, B, C, D, E\}$

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

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

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

(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**

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