

Aula 04 - Busca sem Informação: BFS, DFS, Busca de Custo Uniforme.

Profa. Gabrielly Queiroz

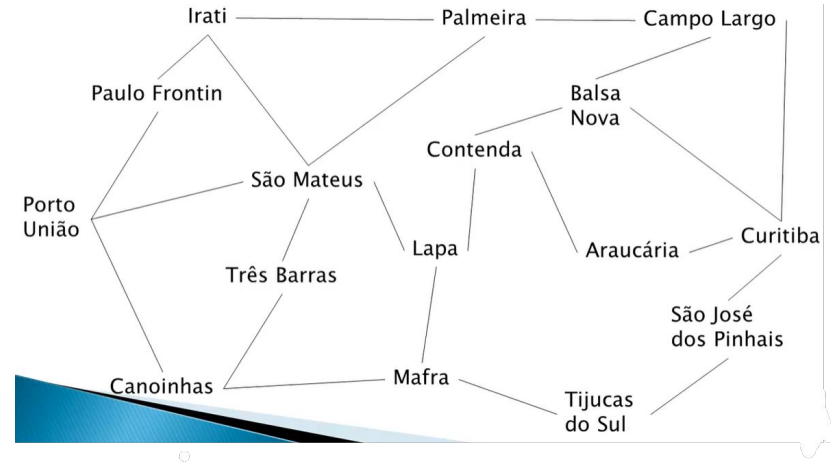
Busca sem informação

A busca sem informação, também chamada de **busca cega**, é um conjunto de estratégias de busca que não utilizam conhecimento adicional sobre o problema além da própria estrutura do grafo. Esses algoritmos exploram os estados do espaço de busca de forma sistemática, sem heurísticas.

- Não leva em conta informações sobre o problema.
- Busca cega/aleatória.

Os três principais algoritmos dessa categoria são:

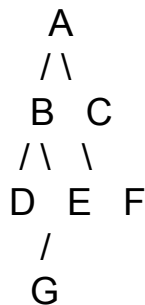
- **BFS (Busca em Largura - Breadth-First Search)**
- **DFS (Busca em Profundidade - Depth-First Search)**
- **Busca de Custo Uniforme (Uniform Cost Search - UCS)**



Representação do Problema

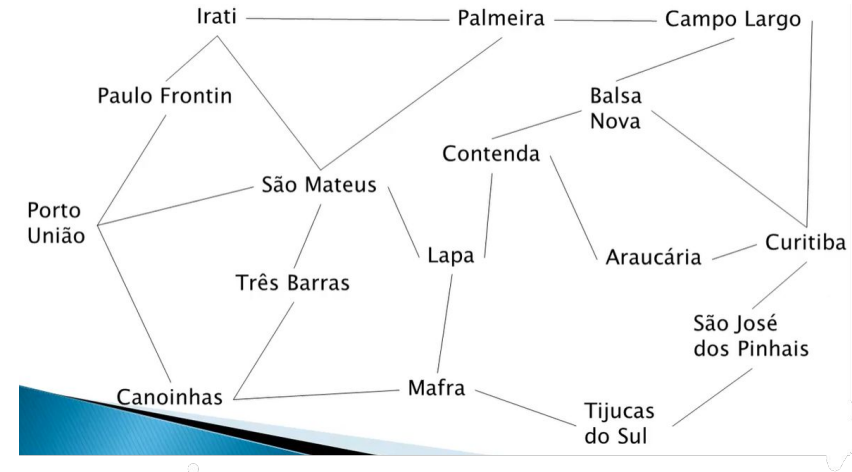
Antes de explorarmos os algoritmos, precisamos entender como representamos um problema de busca. Normalmente, um problema é modelado como um **grafo**, onde:

- Cada **nó** representa um estado do problema.
- Cada **aresta** representa uma transição entre estados.
- O **nó inicial** é o ponto de partida da busca.
- O **nó objetivo** é o estado final desejado.
- O **custo** pode ser associado às transições entre estados.



Algoritmo genérico

- Fronteira do espaço de estados
 - Algoritmo: Atribuir à fronteira (os nós a serem explorados) o estado inicial
1. Selecionar o primeiro nó da fronteira do espaço de estado
 2. Testar se o nó é um estado final (objetivo)
 - a. Caso sim, a busca termina como sucesso
 3. Gerar um novo conjunto de estados
 4. Remover o nó explorado da fronteira
 5. Inserir os nós gerados na fronteira e voltar ao passo 1



Algoritmo genérico

O que muda de um algoritmo para o outro?

A ordem em que as fronteiras são inseridas!

Remover um nó da fronteira (a ordem de remoção define o algoritmo!).

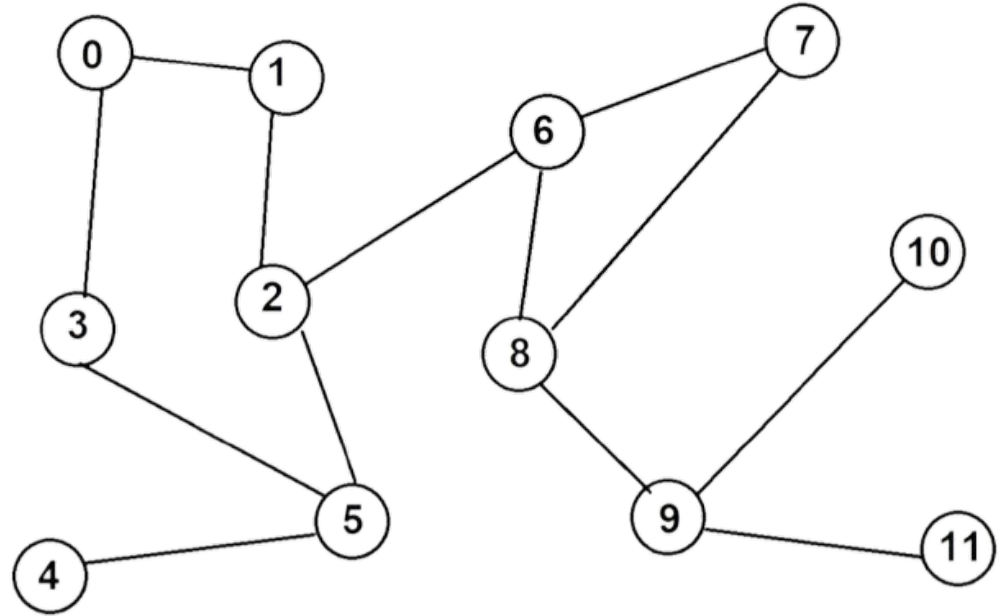
Busca em Largura (BFS - Breadth-First Search)

Explora todos os nós no mesmo nível antes de avançar para níveis mais profundos.

Utiliza uma **fila (FIFO - First In, First Out)** para armazenar os nós a serem explorados.

Garante a solução mais curta (mínimo de passos) se o custo de cada aresta for uniforme.

1. O estado inicial recebe um pai especial (0).
2. Cada nó expandido recebe como pai o nó que o descobriu primeiro (0+1).
3. Para encontrar o melhor caminho, percorremos os pais a partir do nó objetivo até o estado inicial.



Busca em Profundidade (DFS - Depth-First Search)

Explora um caminho completamente antes de retornar e tentar outro.

Utiliza uma **pilha (LIFO - Last In, First Out)**.

Pode seguir um caminho errado por muito tempo antes de corrigir a rota.

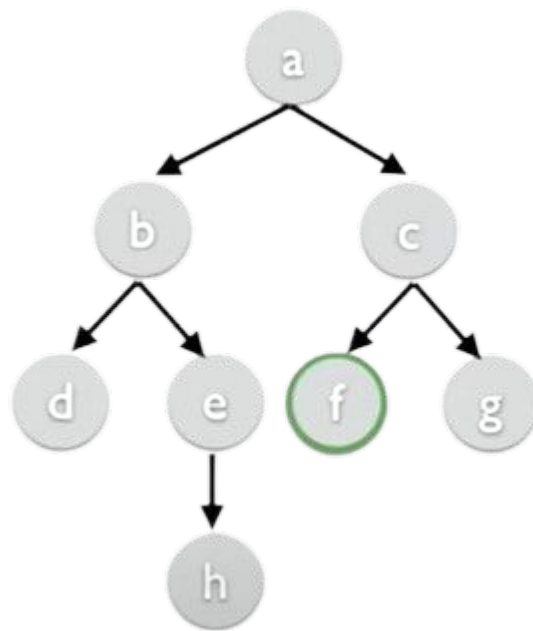
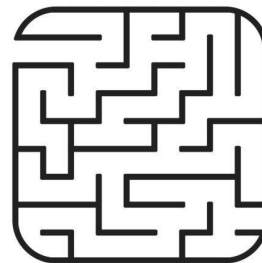
Cada nó expandido recebe como **pai** o nó que o descobriu primeiro, seguindo sempre um **caminho profundo antes de voltar atrás**.

A reconstrução do caminho segue a mesma lógica: ao encontrar o objetivo, percorremos os pais até o estado inicial.

Vantagens e Desvantagens

Usa menos memória que BFS.

Não garante o caminho mais curto.



Busca de Custo Uniforme (Uniform Cost Search - UCS)

Sempre expande o nó com menor custo acumulado.

Utiliza uma **fila de prioridade (heap)** para armazenar os nós.

Considera o custo de cada aresta, diferente do BFS.

Passo a Passo

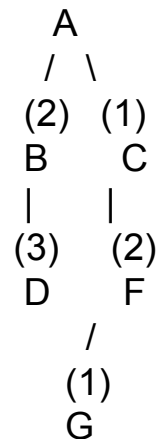
1. Insere o nó inicial na fila de prioridade com custo 0.
2. Remove o nó com menor custo e verifica se é o objetivo.
3. Se não for, adiciona seus vizinhos, somando os custos.
4. Repete o processo até encontrar a solução ou a fila ficar vazia.

Vantagens e Desvantagens

Garante o caminho de menor custo.

Pode ser lento se os custos forem muito altos e variados.

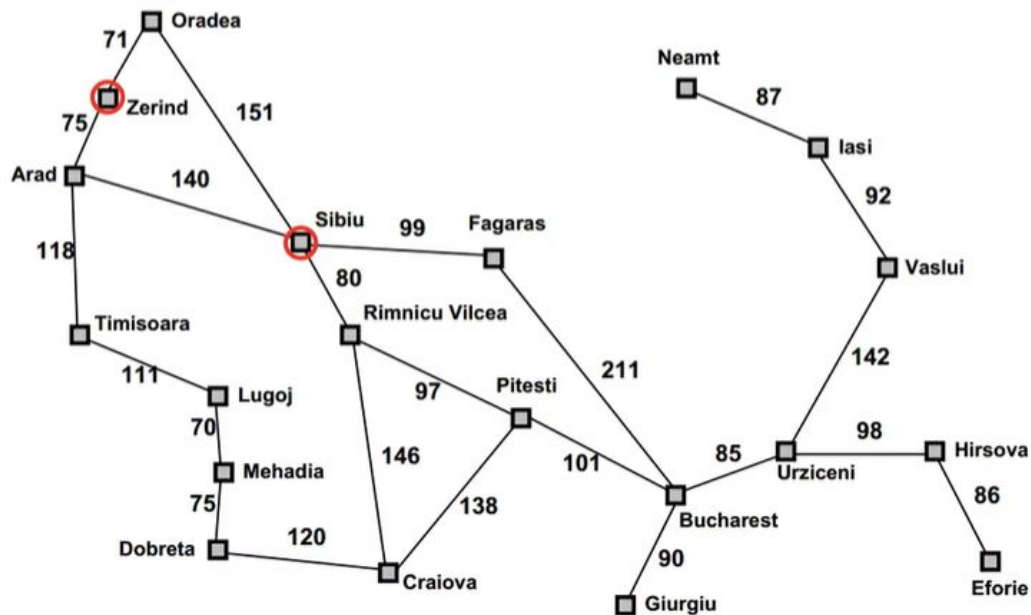
Exemplo - Grafo com custos



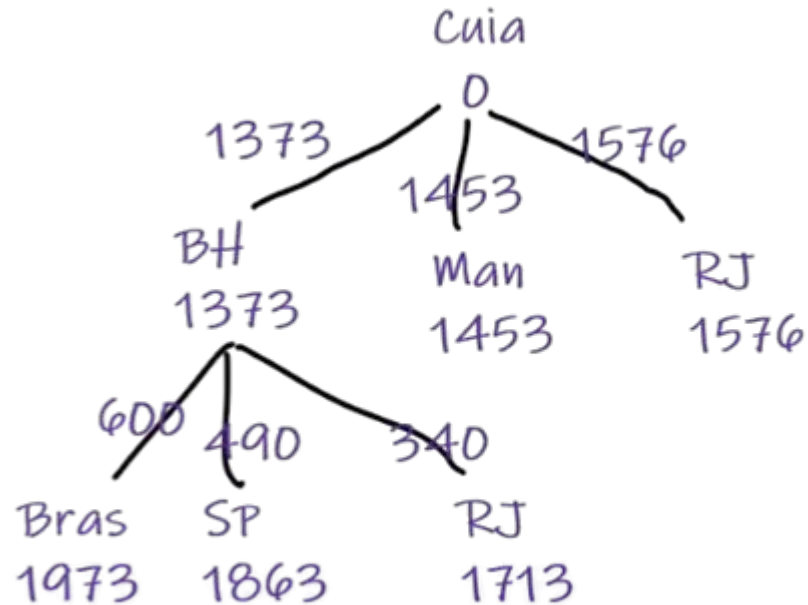
Busca de Custo Uniforme (Uniform Cost Search - UCS)

- Problema: Encontrar caminho de Zerind a Sibiu

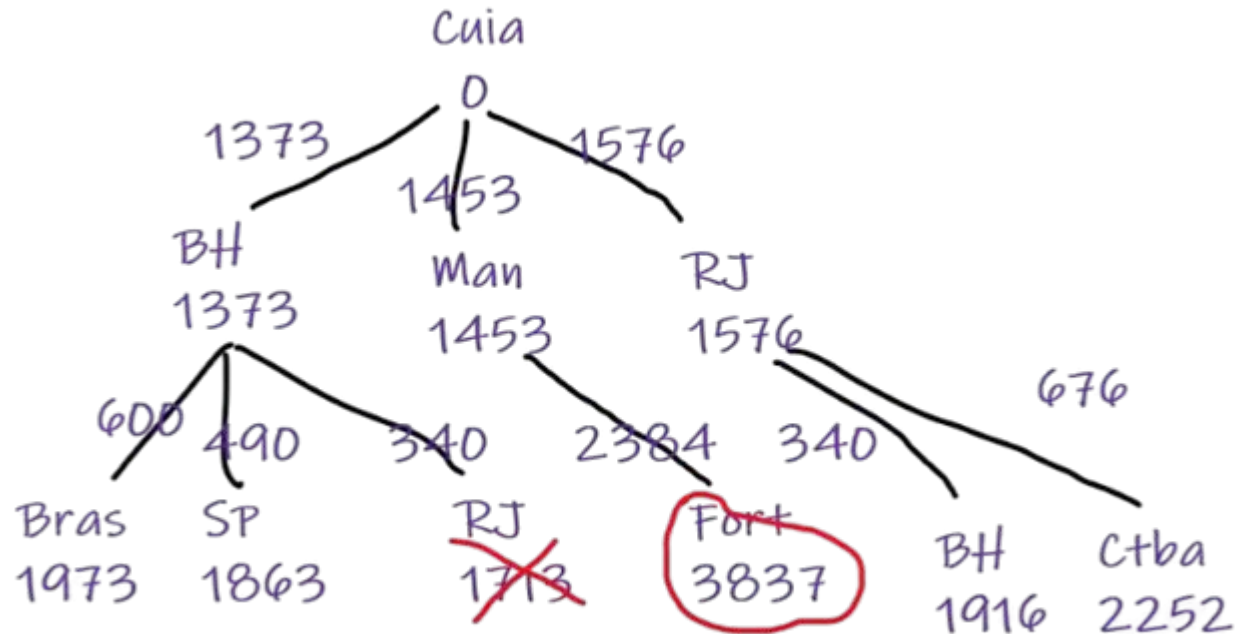
- Estado Inicial: Zerind.
- Estado Objetivo: Sibiu.



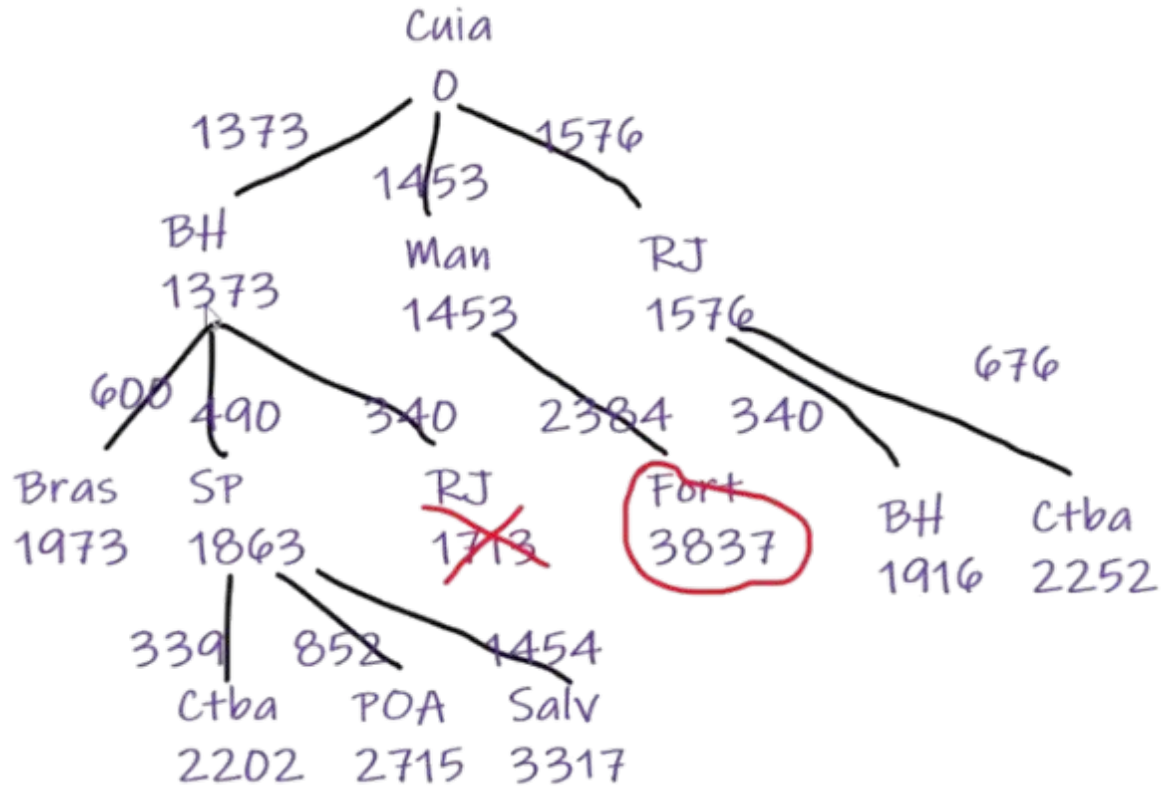
Busca de Custo Uniforme (Uniform Cost Search - UCS)



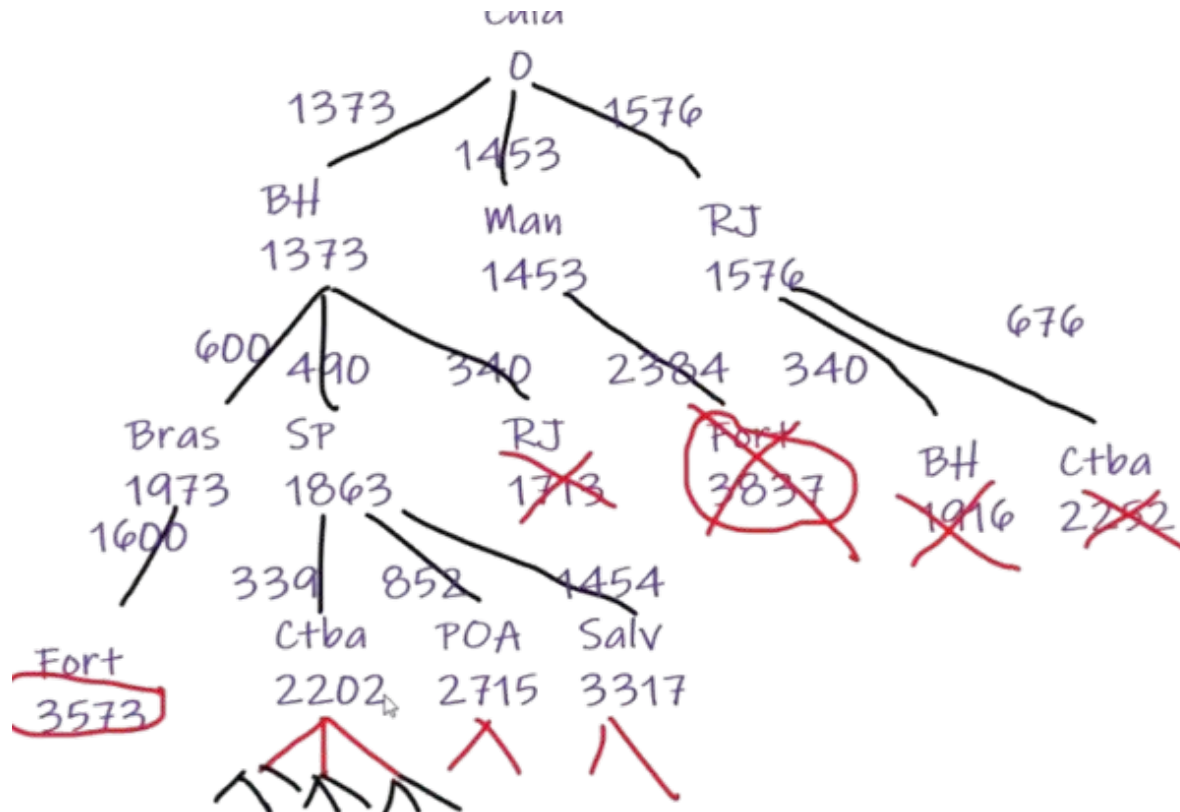
Busca de Custo Uniforme (Uniform Cost Search - UCS)



Busca de Custo Uniforme (Uniform Cost Search - UCS)



Busca de Custo Uniforme (Uniform Cost Search - UCS)



Busca de Custo Uniforme (Uniform Cost Search - UCS)

Quando parar?

Quando todos os custos forem maior que 3573

BFS X DFS X Custo Uniforme x UCS

Critério	BFS (Largura)	DFS (Profundidade)	UCS (Custo Uniforme)
Estrutura	Fila (FIFO)	Pilha (LIFO)	Fila de prioridade
Ordem de Exploração	Nível por nível	Um caminho por vez	Menor custo primeiro
Melhor Caminho?	Se custo uniforme	Não	Sempre
Memória	Alta	Baixa	Alta
Aplicação	Caminho mais curto	Jogos	Caminho de menor custo