

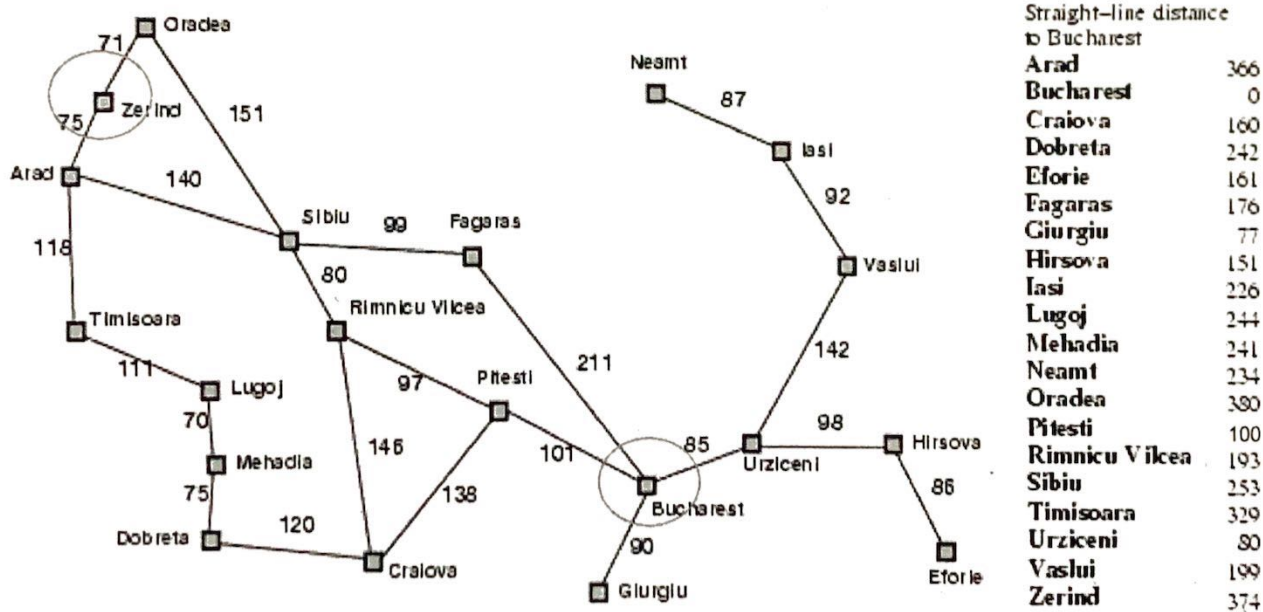
1ª Avaliação Parcial
 Curso: Engenharia da Computação
 Disciplina: Inteligência Computacional
 Prof. Jarbas Joaci de Mesquita Sá Junior
 UFC – Universidade Federal do Ceará

92/11

Nome: _____ Data 14/09/2022

1. Explique como funcionam as buscas em extensão, profundidade e aprofundamento **iterativo em profundidade**. Além disso, descreva as vantagens e desvantagens dessas buscas em termos de **completeza**, **otimalidade**, **complexidade de tempo** (quantidade de nodos gerados) e **complexidade de espaço** (quantidade de nodos armazenados na memória). (4,0 pontos)

2. Considerando o mapa da *Romênia* abaixo e a relação das distâncias em linha reta de cada cidade em relação a *Bucharest* (considerar essas distâncias a heurística $h(n)$ do problema), responda às seguintes solicitações:



Fonte da imagem: adaptado de <http://aima.cs.berkeley.edu>

a) Considerando o estado inicial a cidade de *Zerind* e o estado meta a cidade de *Bucharest*, desenhe as fases de uma busca gulosa pela melhor escolha. (2,0 pontos)

b) Com os mesmos estados inicial e meta do item anterior, desenhe as fases de uma busca A*. (2,0 pontos)

3. O que é uma **heurística admissível**? Explique o conceito de **problema relaxado** e como é possível usá-lo para obter a mencionada heurística? (2,0 pontos)

10)

Busca em extensão

Verifica o nó raiz, se não for o objetivo, expande para os filhos da raiz. Realiza a verificação em todos os filhos para depois expandir para o próximo nível de filhos para verificar. E assim por diante, sempre verificando a camada completa ~~de filhos~~ antes de avançar.

Completeza: É completa se o fator de ramificação b e a profundidade d do nó objetivo não forem finitos.

Otimidade: É ótima se o custo de caminhos forem iguais por nível e maior que zero.

Complexidade de tempo: Exponencial $O(b^{d+1})$

Complexidade de espaço: Exponencial $O(b^{d+1})$

Busca em profundidade

Verifica a raiz, se não for o objetivo, expande a busca para o filho da ~~ra~~ borda, se este não for o objetivo, expande-se os filhos deste e verifica o nó da borda novamente. E assim por diante até atingir a profundidade da árvore. Se atingir a profundidade da árvore e não for o objetivo, a busca retorna para os nós que têm filhos ainda não analisados.

O nó e/ou ~~seu~~ seu ramo são excluídos se não contarem o objetivo.

Completeza: Não é completa se a árvore for muito longa ou infinita, pois o objetivo pode estar em outro ramo ~~o~~ próximo à raiz e a busca estará presa no ramo longo ou infinito.

Otimidade: Não é ótima pois não garante encontrar o objetivo com o menor custo possível. Podendo até nem encontrar.

Continuação →

→

Complexidade de tempo: Exponencial $O(b^n)$

Complexidade de espaço: Linear $O(bm)$

Busca em profundidade iterativa:

Combina as vantagens da busca em extensão e a da busca em profundidade. Se estabelece um limite inicial que aumenta gradualmente a cada iteração de busca. A cada iteração verifica-se desde a raiz até a camada do limite atual, verificando a camada completa desse limite. Repete-se o processo até encontrar o objetivo.

Completeness: É completa se o fator de ramificação for finito.

Optimality: É ótima se os custos de caminho forem iguais por nível e maior que zero.

Complexidade de tempo: Exponencial $O(b^d)$

Complexidade de espaço: Linear $O(bd)$

3º) Heurística

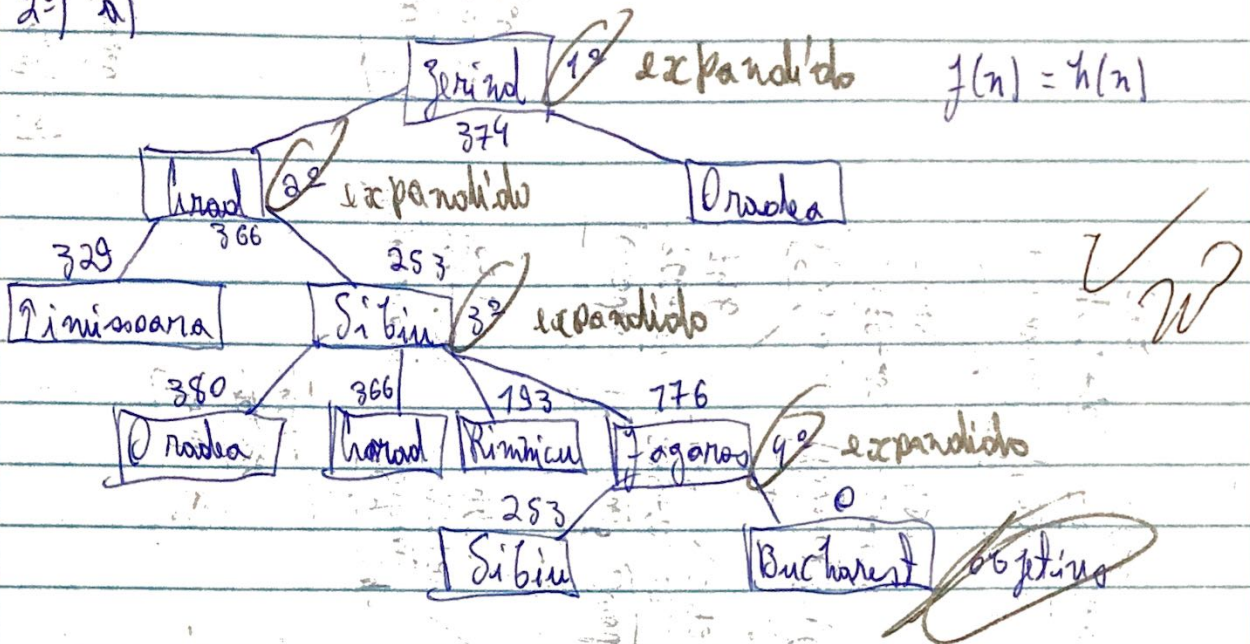
Uma heurística admissível é aquela que não superestima o custo até o objetivo. É ótima por natureza.

O problema relaxado é um problema com poucas restrições.

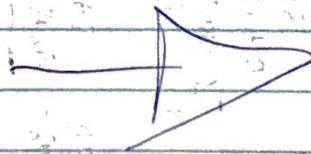
Um problema relaxado tem heurística admissível pois terá custo ótimo.

✓

2º a)



Zerind → Arad → Sibiu → Fagaras → Bucharest.



Continuação da
questão

2.01 b)

$$f(m) = g(m) + h(n)$$

