

Inteligência Artificial

Tópico 03 - Parte 01

Resolução de Problemas por Buscas

Profa. Dra. Priscila Tiemi Maeda Saito
✉ priscilasaito@ufscar.br

1 Resolução de Problemas por Busca

- Formulação de Objetivo
- Formulação do Problema

Motivação

Problema - Estradas Escuras¹

**BEE 1152**

GRAFOS | NÍVEL 5 | + 5.5 PONTOS | TEMPO LIMITE BASE: 3 SEGUNDOS | LIMITE DE MEMÓRIA: 200 MB

beecrowd | 1152

Estradas Escuras

Univeristy of Ulm Local Contest  Alemanha

Timelimit: 3

Nestes dias se pensa muito em economia, mesmo em Byteland. Para reduzir custos operacionais, o governo de Byteland decidiu otimizar a iluminação das estradas. Até agora, todas as rotas eram iluminadas durante toda noite, o que custava 1 Dólar Byteland por metro a cada dia. Para economizar, eles decidiram não iluminar mais todas as estradas e desligar a iluminação de algumas delas. Para ter certeza que os habitantes de Byteland continuem a se sentirem seguros, eles querem otimizar o sistema de tal forma que após desligar a iluminação de algumas estradas à noite, sempre existirá algum caminho iluminado de qualquer junção de Byteland para qualquer outra junção.

Qual é a quantidade máxima de dinheiro que o governo de Byteland pode economizar, sem fazer os seus habitantes sentirem-se inseguros?

Entrada

A entrada contém vários casos de teste. Cada caso de teste inicia com dois números m ($1 \leq m \leq 200000$) e n ($m-1 \leq n \leq 200000$), que são o número de junções de Byteland e o número de estradas em Byteland, respectivamente. Seguem n conjuntos de três valores inteiros, x , y e z , especificando qual será a estrada bidirecional entre x e y com z metros ($0 \leq x, y < m$ e $x \neq y$).

A entrada termina com $m=n=0$. O grafo especificado em cada caso de teste é conectado. O tamanho total de todas as estradas em cada caso de teste é menor do que 2^{31} .

Saída

Para cada caso de teste imprima uma linha contendo a máxima quantidade diária de dólares de Byteland que o governo pode economizar.

Exemplo de Entrada	Exemplo de Saída
7 11 0 1 7 0 3 5 1 2 8 1 3 9 1 4 7 2 4 5 3 4 15 3 5 6 4 5 8 4 6 9 5 6 11 0 0	51

¹[https://www.beecrowd.com.br/judge/pt/problems/view/1152\(antigoURIOnlineJudge\)](https://www.beecrowd.com.br/judge/pt/problems/view/1152(antigoURIOnlineJudge))

Motivação

Problema - Estradas Escuras ²

- Nestes dias se pensa muito em economia, mesmo em *Byteland*.
 - Para reduzir custos operacionais, o governo de *Byteland* decidiu **otimizar a iluminação das estradas**.
 - Até agora, todas as rotas eram iluminadas durante toda noite, o que custava 1 Dólar *Byteland* por metro a cada dia.
 - Para economizar, eles decidiram não iluminar mais todas as estradas e desligar a iluminação de algumas delas.
 - Para ter certeza que os habitantes de *Byteland* continuem a se sentirem seguros, eles querem otimizar o sistema de tal forma que após desligar a iluminação de algumas estradas à noite, sempre existirá algum caminho iluminado de qualquer junção de *Byteland* para qualquer outra junção.
-
- Qual é a quantidade máxima de dinheiro que o governo de *Byteland* pode economizar, sem fazer os seus habitantes sentirem-se inseguros?

²[https://www.beecrowd.com.br/judge/pt/problems/view/1152\(antigoURIOnlineJudge\)](https://www.beecrowd.com.br/judge/pt/problems/view/1152(antigoURIOnlineJudge))

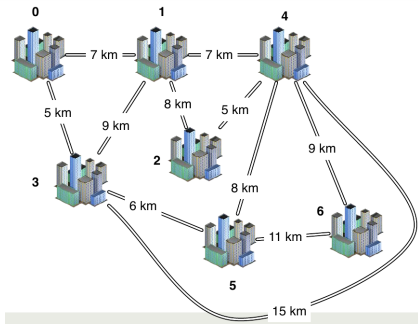
Motivação

- Como representar computacionalmente as estradas e os cruzamentos entre as mesmas?
- E para resolver o problema?

Motivação

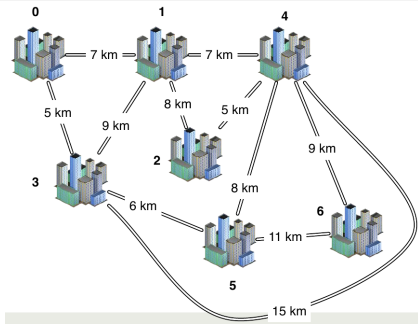
- Como representar computacionalmente as estradas e os cruzamentos entre as mesmas?

• E para resolver o problema?



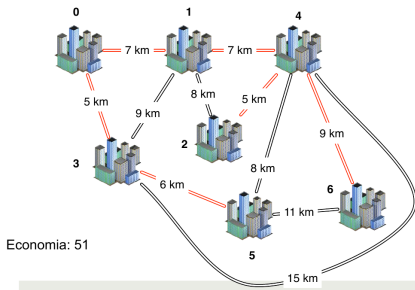
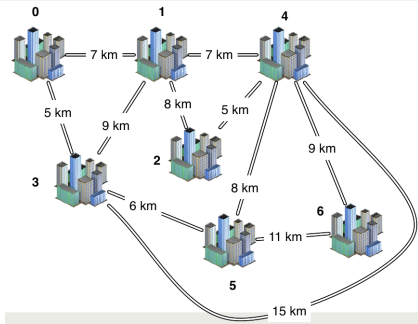
Motivação

- Como representar computacionalmente as estradas e os cruzamentos entre as mesmas?
- E para resolver o problema?



Motivação

- Como representar computacionalmente as estradas e os cruzamentos entre as mesmas?
- E para resolver o problema?



Grafos/redes

- Estruturas importantes e recorrentes com aplicação em diversas áreas do conhecimento
- Centenas de problemas computacionais usam **grafos** com sucesso
- Representam arbitrárias relações entre elementos/entidades/objetos
- Muitas aplicações em computação necessitam considerar conjunto de conexões entre pares de objetos
 - ▶ existe um caminho para ir de um objeto a outro seguindo as conexões?
 - ▶ qual é a menor distância entre um par de objetos?
 - ▶ quantos objetos podem ser alcançados a partir de um determinado objeto?
- Criação de heurísticas ótimas/sub-rotinas para realizar **busca de padrões em redes reais**

Aplicações

- O que podemos modelar por meio de grafos?
 - ▶ Redes Tecnológicas
 - ▶ Redes Sociais
 - ▶ Redes de Informações
 - ▶ Redes Biológicas
- Exemplos de aplicações de problemas práticos que podem ser resolvidos por meio de uma modelagem em grafos?

Aplicações

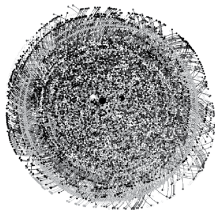
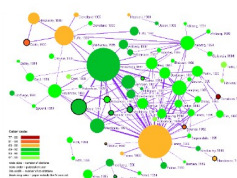
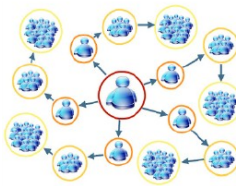
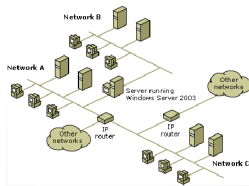


Figura 1.2 – Rede complexa composta de genes relacionados à doença de Alzheimer e seus relacionamentos.

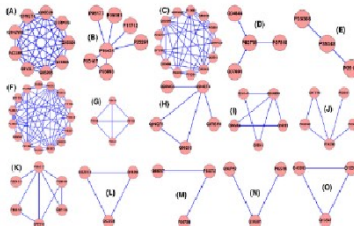


Fig. 4. 15 sub-graphs of the human protein-protein interaction network. The nodes are colored by the number of interactions they have with other proteins.

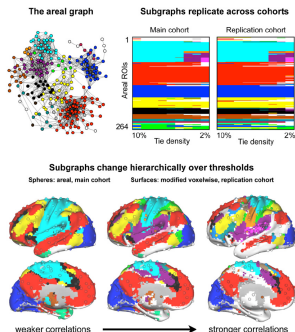
Interações de Proteínas³

³Zhan Chao Li et al. Identification of human protein complexes from local sub-graphs of protein-protein interaction network based on random forest with topological structure features, Analytica chimica acta, 2012.

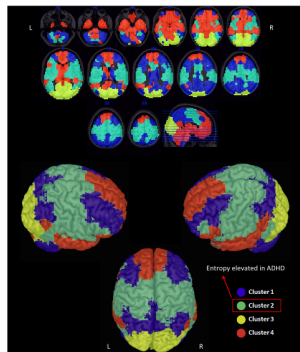
Aplicações

• Redes de Funções Cerebrais

- ▶ mapeamento de regiões cerebrais em grafos
- ▶ análise da condição cerebral de indivíduos com doenças (Alzheimer, Esquizofrenia, Esclerose múltipla)



4



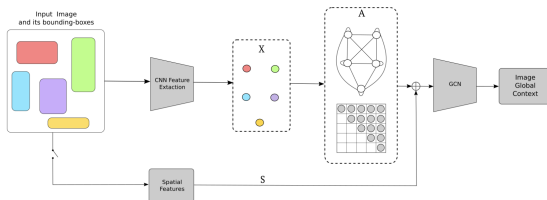
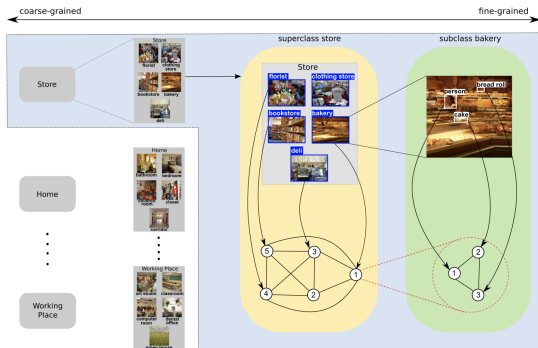
5

Fig. 5. Brain mapping of the functional MNI clusters. The edges of cluster 2 (significantly reduced in ADHD) are occupied by typically developing children. This cluster consists of bilateral pre and postcentral cortex, superior temporal gyrus, and inferior frontal gyrus.

⁴ JD Power et al. Functional network organization of the human brain. Neuron, 2011.

⁵ JR Sato et al. Measuring network's entropy in ADHD: A new approach to investigate neuropsychiatric disorders. Neuroimage. 2013.

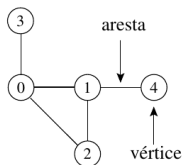
Aplicações



6

Grafos

- Conjunto de vértices (entidades) conectados por arestas (relacionamento)
- Notação: $\mathbf{G} = (\mathbf{V}, \mathbf{E})$
 - ▶ conjunto \mathbf{V} com n vértices, ou nós
 - ★ $\{v_1, v_2, v_3, \dots, v_n\}$
 - ▶ conjunto \mathbf{E} com m arestas, ou arcos
 - ★ $\{e_1, e_2, e_3, \dots, e_m\}$

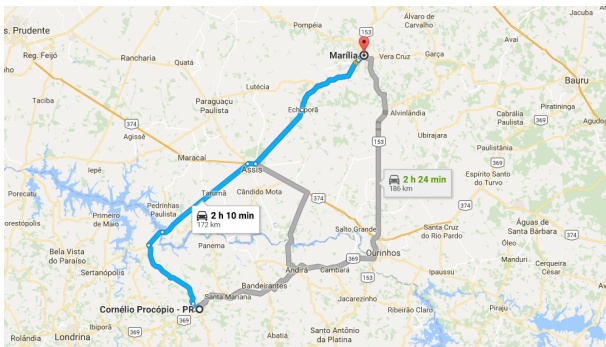


Motivação

- Estratégia de busca é uma das mais **poderosas** abordagens para resolução de problemas em IA
- Explora sistematicamente as **alternativas**
- Encontra a sequência de **passos para uma solução**

Exemplo de Problema

- Viajar de Cornélio Procópio a Marília
 - ▶ adotando a menor rota possível
 - ▶ 172 km - aprox. 2 horas e 10 minutos - Google Maps



Resolução de Problemas

- Primeiro passo: formular objetivos
 - ▶ o que deve ser alcançado
 - ★ ir a Marília
 - ▶ baseado em situação atual
 - ★ estamos em Cornélio Procópio
 - ▶ e medida de desempenho
 - ★ quero seguir a **menor rota** rodoviária possível
- Objetivo pode ser visto como um **estado**
 - ▶ entre possíveis estados do problema
 - ★ Ex.: estado: estar em alguma cidade

Resolução de Problemas

- Podem haver estados intermediários no “caminho” da solução
 - ▶ Ex.: há duas estradas saindo de Cornélio Procópio (CP)
 - ★ CP → Assis
 - ★ CP → Santa Mariana
- Descobrir que sequência de ações levará ao objetivo

Que cidades percorrer para chegar a Marília?

Resolução de Problemas

- Segundo passo: formular o problema
 - ▶ processo de decidir que ações e estados devem ser considerados, dado um objetivo
 - ▶ No exemplo:
 - ★ Ações = direções
 - ★ Estados = cidades percorridas
 - ★ Objetivo = chegar a Marília a partir de Cornélio Procópio usando menor rota

Resolução de Problemas

- Terceiro passo: buscar solução

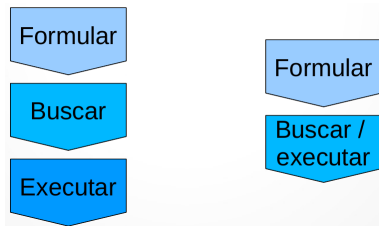
- ▶ processo de procurar por sequência de ações que alcançam o objetivo
- ▶ **algoritmo de busca**
 - ★ entrada = problema
 - ★ saída = sequência de ações para chegar à solução

Resolução de Problemas

- “Uma vez a solução é encontrada, as ações recomendadas podem ser executadas”
- Execução
 - ▶ no exemplo consiste em seguir a rota de cidades recomendada
 - ▶ execução pode ser intercalada com a busca
 - ★ robô percebendo o ambiente e então navegando

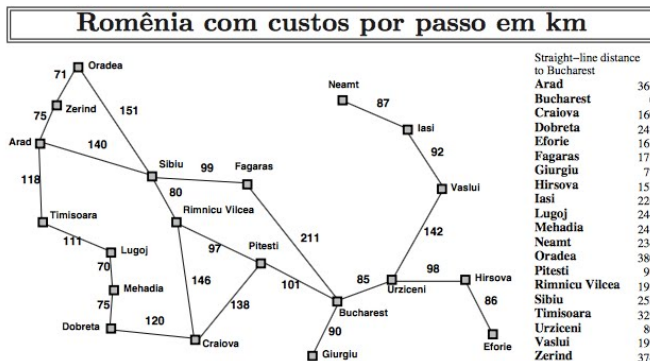
Resolução de Problemas

- Três etapas principais:
 - ▶ formulação do problema
 - ▶ busca de soluções
 - ▶ execução da solução



Outro Exemplo

- Para o problema da Romênia



Problema

Definido por cinco componentes:

- 1 Estado inicial
- 2 Ações/operadores possíveis para gerar novo estado
- 3 Modelo de transição
- 4 Teste de objetivo
- 5 Uma função de custo de caminho

Problema

- Estado inicial
 - ▶ Ex.: Em(Arad)
- Ações/operadores possíveis para gerar novo estado
 - ▶ Ex.: $\text{Em(Arad)} = \{\text{Ir(Sibiu)}, \text{Ir(Timisoara)}, \text{Ir(Zerind)}\}$
 - ▶ Estado inicial e operadores definem **espaço de estados**
 - ★ conjunto de todos os estados acessíveis a partir do estado inicial
 - ★ forma um grafo

Problema

- Modelo de transição
 - ▶ descrição do que cada ação faz
 - ▶ Ex.: $\text{RESULTADO}(\text{Em}(\text{Arad}), \text{Ir}(\text{Zerind})) = \text{Em}(\text{Zerind})$
- Teste de objetivo
 - ▶ determinar se um dado estado é um estado objetivo
 - ▶ No ex., o objetivo é o conjunto unitário $\{\text{Em}(\text{Bucareste})\}$
- Uma função de custo de caminho
 - ▶ atribui custo numérico a cada caminho no grafo
 - ▶ caminho é uma sequência de estados conectados
 - ▶ geralmente reflete medida de desempenho
 - ★ no ex., menor caminho em km

Solução

- Uma **solução** para um problema é um caminho desde o estado inicial até o estado objetivo
 - ▶ **solução ótima tem menor custo de caminho entre todas as possíveis soluções**

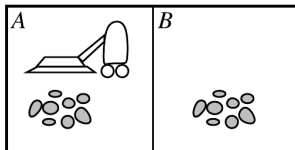
Como Formular um Problema?

Como escolher estados e ações?

- Abstrair detalhes irrelevantes
 - ▶ na formulação do problema (ações e estados)
 - ▶ nas funções de custo de caminho e de busca
- Exemplo: dirigir de Cornélio Procópio a Marília
 - ▶ não é de interesse:
 - ★ número de passageiros
 - ★ o que toca no rádio (estado)
 - ★ cidades sem acesso rodoviário (estado)
 - ★ o que se come e bebe dentro do carro (ações)
 - ★ número de postos policiais no caminho (custo de caminho)
 - ▶ é tarefa do projetista fazer as boas escolhas

Exemplo Problema 1

- Aspirador de pó
 - ▶ percorre quadrados e verifica se tem sujeira para limpar



- Operações possíveis:
 - ▶ mover para a direita
 - ▶ mover para a esquerda
 - ▶ aspirar pó
 - ▶ não fazer nada

Quais as etapas para a resolução do problema?

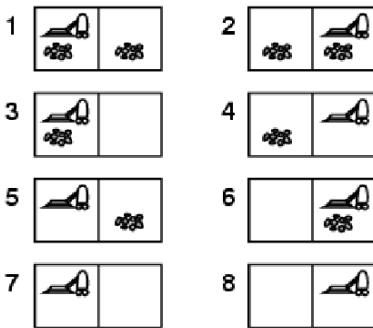
Como formular o problema?

Aspirador de Pó

- Formulação do problema

- ▶ estados:

- ★ 2 quadrados, contendo ou não sujeira
 - ★ 8 estados possíveis



Aspirador de Pó

- Formulação do problema:

- ▶ estado inicial:

- ★ qualquer um dos estados possíveis
 - ★ ex.: estado 5



- ▶ ações:

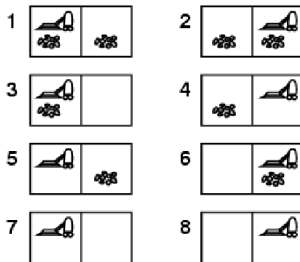
- ★ direita
 - ★ esquerda
 - ★ limpar

Aspirador de Pó

- Formulação do problema:
 - ▶ teste de objetivo:
 - ★ verificar se quadrados estão limpos
 - ▶ custo de caminho:
 - ★ cada passo custa 1
 - ★ custo do caminho é o número de passos do caminho

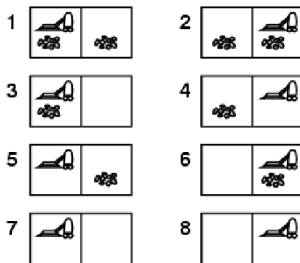
Aspirador de Pó

- O estado inicial é um dos possíveis estados: {1, 2, 3, 4, 5, 6, 7, 8}
- Executar a ação **moverDireita** sempre vai levar o agente para um dos estados:
 - ▶ ?
- A sequência de ações [MoverDireita, Limpar] sempre vai levar o agente para um dos estados:
 - ▶ ?
- A sequência de ações [MoverDireita, Limpar, MoverEsquerda, Limpar] sempre vai levar o agente para o estado:
 - ▶ ?



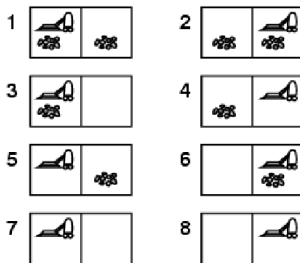
Aspirador de Pó

- O estado inicial é um dos possíveis estados: {1, 2, 3, 4, 5, 6, 7, 8}
- Executar a ação **moverDireita** sempre vai levar o agente para um dos estados:
 - ▶ {2, 4, 6, 8}
- A sequência de ações [MoverDireita, Limpar] sempre vai levar o agente para um dos estados:
 - ▶ ?
- A sequência de ações [MoverDireita, Limpar, MoverEsquerda, Limpar] sempre vai levar o agente para o estado:
 - ▶ ?



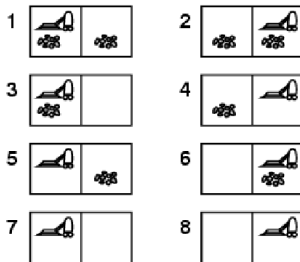
Aspirador de Pó

- O estado inicial é um dos possíveis estados: {1, 2, 3, 4, 5, 6, 7, 8}
- Executar a ação **moverDireita** sempre vai levar o agente para um dos estados:
 - ▶ {2, 4, 6, 8}
- A sequência de ações [MoverDireita, Limpar] sempre vai levar o agente para um dos estados:
 - ▶ {4, 8}
- A sequência de ações [MoverDireita, Limpar, MoverEsquerda, Limpar] sempre vai levar o agente para o estado:
 - ▶ ?



Aspirador de Pó

- O estado inicial é um dos possíveis estados: {1, 2, 3, 4, 5, 6, 7, 8}
- Executar a ação **moverDireita** sempre vai levar o agente para um dos estados:
 - ▶ {2, 4, 6, 8}
- A sequência de ações [MoverDireita, Limpar] sempre vai levar o agente para um dos estados:
 - ▶ {4, 8}
- A sequência de ações [MoverDireita, Limpar, MoverEsquerda, Limpar] sempre vai levar o agente para o estado:
 - ▶ 7



Exemplo Problema 2

• Quebra-cabeça de 8 peças

- ▶ tabuleiro 3x3 com 8 peças numeradas e um espaço vazio
- ▶ uma peça pode deslizar para o espaço
- ▶ exemplo:

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

- **Formulação do problema:** um estado especifica a posição de cada peça e do espaço vazio nos 9 quadrados
 - ▶ **estado inicial:** qualquer estado
 - ▶ **ações:** espaço vazio se desloca para esquerda, direita, cima ou baixo
 - ▶ **modelo de transição:** dado um estado e ação, ele devolve estado resultante (move 6 para o lado esquerdo)
 - ▶ **teste de objetivo:** o estado é o final?
 - ▶ **custo do caminho:** cada passo custa 1 e custo do caminho = # de passos no caminho

Problema dos quebra-cabeças deslizantes

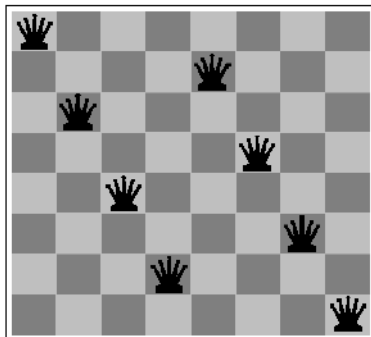
- NP-completo
 - ▶ ainda não existem algoritmos determinísticos que encontram a solução em tempo polinomiais
 - ★ 8 peças: 181440 estados possíveis
 - ★ 15 peças: 1.3 trilhão de estados
 - ★ 24 peças: 10^{25} estados

Exemplo Problema 3

● Problema das 8 rainhas

- ▶ posicionar 8 rainhas em um tabuleiro de xadrez de forma que nenhuma rainha ataque outra

Estado Objetivo?

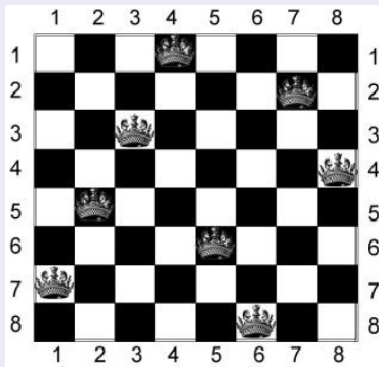


Exemplo Problema 3

- **Problema das 8 rainhas**

- ▶ posicionar 8 rainhas em um tabuleiro de xadrez de forma que nenhuma rainha ataque outra

Estado Objetivo?



8 rainhas

- **Formulação incremental:** adiciona rainhas ao tabuleiro
 - ▶ **estados**^{7 8 9}: qualquer disposição de 0 a 8 rainhas no tabuleiro
 - ▶ **estado inicial:** nenhuma rainha no tabuleiro
 - ▶ **ações:** colocar uma rainha qualquer em um espaço vazio
 - ▶ **teste de objetivo:** 8 rainhas no tabuleiro e nenhuma é atacada
 - ▶ **custo:** não é de interesse, pois apenas o estado final é importante

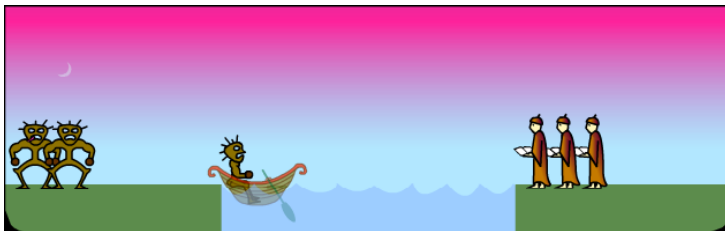
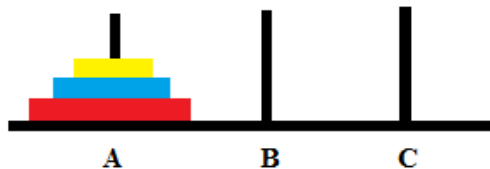
⁷força bruta -> $C_{64,8} = 64! / (8! * (64-8)!) = 64! / (8! * 56!) = 4426165368$ (Espaço de busca grande demais!)

⁸redução do espaço de busca -> $8^8 = 16777216$ (colunas diferentes)

⁹redução do espaço de busca -> $8! = 40320$ (linhas diferentes)

Outros Problemas

- Torre de Hanoi
- Canibais e missionários



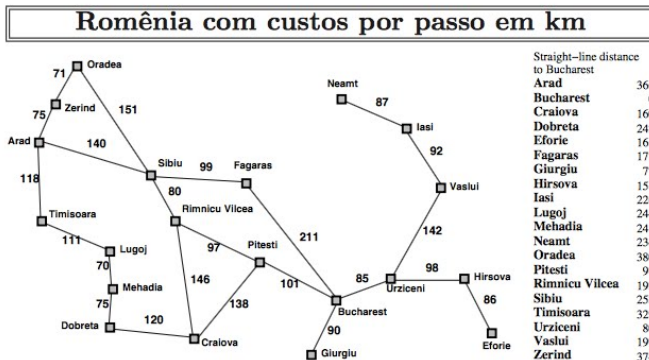
Busca por Soluções

Resolução dos problemas após formulação

- por meio de busca no espaço de estados
 - ▶ sequência de ações possíveis formam **árvore de busca**:
 - ★ com estado inicial na raiz
 - ★ ramos são as ações
 - ★ nós são os estados no espaço de estados do problema

Exemplo

- Para o problema da Romênia



Exemplo

- (a) Estado inicial

Arad

Raiz da árvore de busca

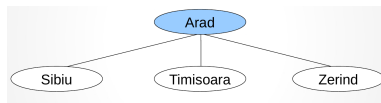
É objetivo?

↓ Não

Expandir estado atual
(gerar novos estados)

Exemplo

- (b) Expandindo Arad



Qual escolher para futuras considerações?

Essência da busca: seguir uma opção e deixar as outras reservadas para mais tarde, no caso da primeira não levar a uma solução

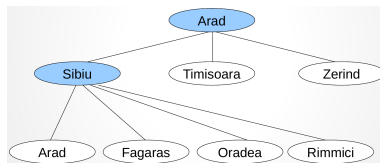
Exemplo

- (c) Supor escolha de Sibiu



Exemplo

- (c) Expandindo Sibiu



Pode escolher entre quaisquer estados ainda não visitados

- Continua-se a escolher, testar e expandir até
 - ▶ encontrar solução ou
 - ▶ não existirem mais estados a serem visitados
- A estratégia de escolha de estado a visitar e expandir é determinada pela **estratégia de busca**

- Algoritmo geral de busca em árvore

```
função BUSCA-EM-ÁRVORE(problema, estratégia) retorna solução  
ou falha  
  iniciar a árvore de busca com o estado inicial  
  repita  
    se não existe nenhum candidato para expansão então  
      retornar falha  
    escolher nó para expansão de acordo com estratégia  
    se o nó contém um estado objetivo então  
      retornar a solução  
    senão  
      expandir o nó  
      adicionar os nós resultantes à árvore de busca
```

Árvore de Busca

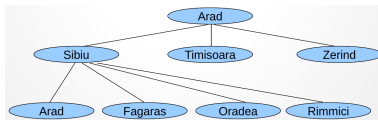
- Os nós da árvore podem guardar mais informação do que apenas o estado: estrutura de dados com pelo menos 5 componentes:
 - 1 o estado correspondente
 - 2 o seu nó pai
 - 3 o operador aplicado para gerar o nó (a partir do pai)
 - 4 a profundidade do nó
 - 5 o custo do nó (desde a raiz)
 - 6 os nós filhos

Árvore de Busca

- Exemplo:

- ▶ Nó Sibiu

- ★ pai = Arad
 - ★ operador = ir de Arad a Sibiu
 - ★ profundidade = 1
 - ★ custo = x km
 - ★ filhos = {Arad, Fagaras, Oradea, Rimmici}



Referências e Leituras Complementares

- Cap. 03 - livro Russel e Norvig
- Cap. 03 - livro Ben Coppin