Introdução a Inteligência Artificial

Aula 3

Prof. Lucas Cambuim

Busca de Soluções

Capítulo 3 – Russell & Norvig

Ao final desta aula a gente deve...

- Compreender o que é um problema de busca em IA
- Ser capaz de formulá-lo
- Conhecer algumas aplicações
- Entender como buscar a solução do problema
  - Busca cega
  - Busca heurística

Problema: Jarros

- Dados uma bica d`agua, um jarro de capacidade 3 litros e um jarro de capacidade 4 litros (ambos vazios). Como obter 2 litros no jarro de 4?

Solução de problemas por meio de busca

- Desenvolver programas, não com os passos de solução de um problema, mas que produzam estes passos;
- Recebe um problema e retorna uma solução

Um problema de busca em IA pode ser definido em termos de... Algumas definições básicas (1/2)

- Um espaço de estados possíveis, incluindo:
  - um estado inicial
    - Em (Recife)
    - Estar (pobre)
  - um ou mais estados finais => objetivo
    - Em (João Pessoa)
    - Estar (rico)
  - Espaço de Estados:
    - conjunto de todos os estados alcançáveis a partir do estado inicial por qualquer sequência de ações.
    - pode ser representado como uma **árvore** onde os estados são nós e as operações são arcos.
  - Ex., dirigir de Recife a João Pessoa
    - Espaço de estados: todas as cidades da região

Um problema de busca em IA pode ser definido em termos de... Algumas definições básicas (1/2)

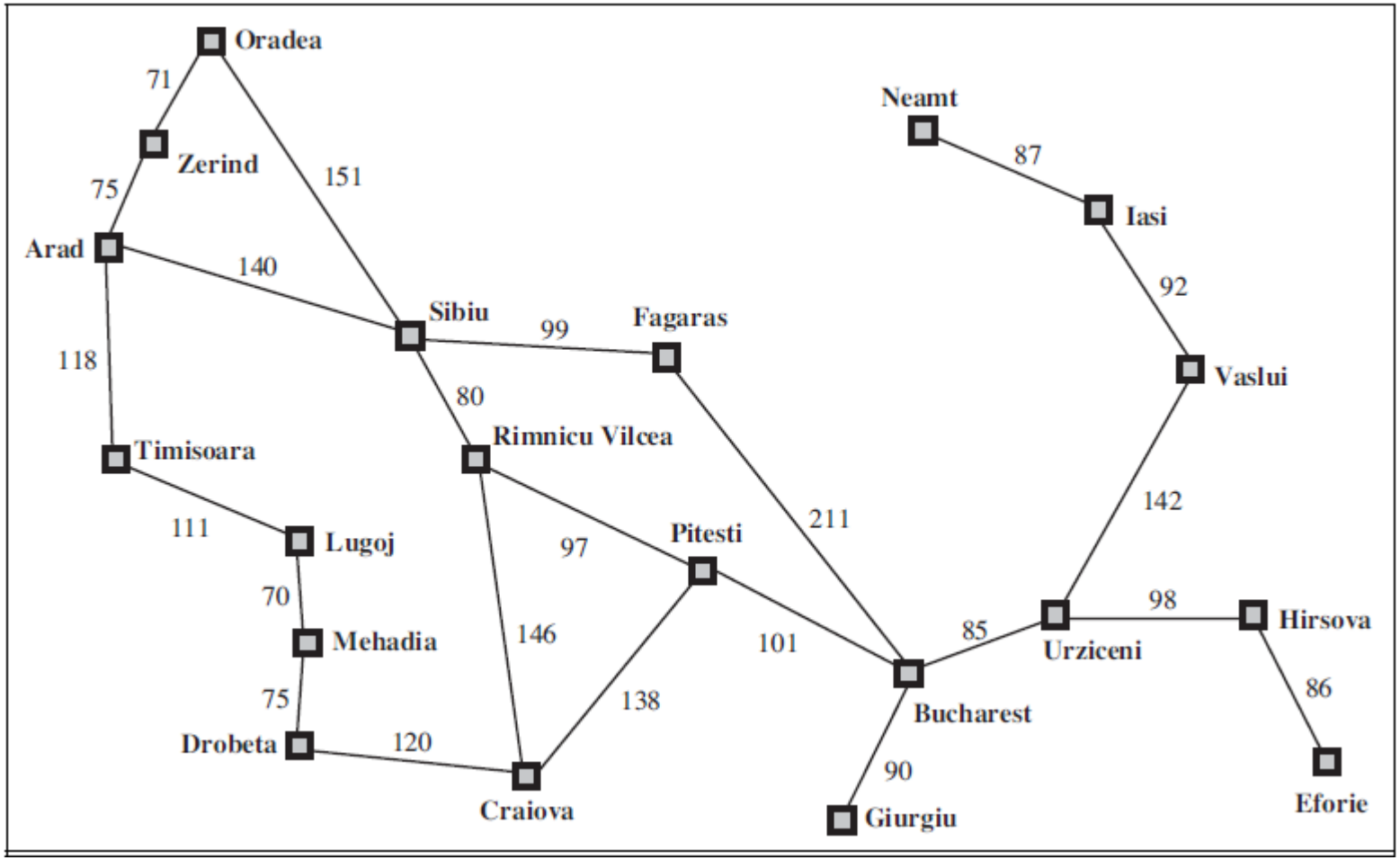
- Um conjunto de ações (ou operadores) que permitem passar de um estado a outro
  - Ex., dirigir de Recife a João Pessoa
    - Ações: dirigir de uma cidade a outra na região
    - E.g. Dirigir (Recife, Abreu e Lima)
  - Ficar rico
    - Jogar(megasena).
  - Os estados podem não "estar lá" concretamente.
    - No caso do problema de dirigir… as cidades estão lá.
    - No caso de ficar rico … não necessariamente.

Um problema de busca em IA pode ser definido em termos de...

- Um estado inicial
- Um conjunto de ações (ou operadores)
  - que permitem passar de um estado a outro
  - os estados podem não “estar lá” concretamente.
    - No caso do problema de dirigir… as cidades estão lá
    - No caso de ficar rico… não necessariamente.
- Um teste de término
  - Verifica se um dado estado é o objetivo
  - Objetivo => um ou mais estados finais
    - propriedade abstrata (em intenção)
      - ex., condição de xeque-mate no Xadrez
    - conjunto de estados finais do problema (em extensão)
      - ex., estar em João Pessoa

Um problema de busca em IA pode ser definido em termos de...

- Custo de caminho
  - Função que associa um custo a cada caminho possível
  - Um caminho é uma sequência de estados conectados por ações possíveis.
  - Cada ação tem um custo associado
    - O custo de dirigir de Recife a Abreu e Lima, por exemplo, poderia ser a distância entre as duas cidades.


Oradea
71
Neamt
Zerind
87
151
75
Iasi
Arad
140
92
Sibiu
99
Fagaras
118
Vaslui
80
Rimnicu Vilcea
Timisoara
142
111
211
Pitesti
Lugoj
97
98
70
Hirsova
Mehadia
146
85
101
Urziceni
75
86
138
Bucharest
Drobeta
120
90
Craiova
Eforie
Giurgiu

Algumas definições

- Solução
  - Caminho (seqüência de ações) que leva do estado inicial a um estado final (objetivo).
  - Cuidado! A solução **não** é o estado final!
    - **Solução** é um caminho desde o estado inicial até o estado final;
    - **Solução ótima**: tem o menor custo de caminho dentre todas as soluções existentes

Solucionando o problema: formulação, busca e execução

- Formulação do problema e do objetivo (manual)
  - quais são os estados e as ações a considerar?
  - qual é (e como representar) o objetivo?
    - Em extensão ou intensão?
- Busca (processo automático)
  - processo que gera/analisa seqüências de ações para alcançar um objetivo
  - solução = caminho entre estado inicial e estado final.
- Execução (manual ou automática)

Custo da Busca

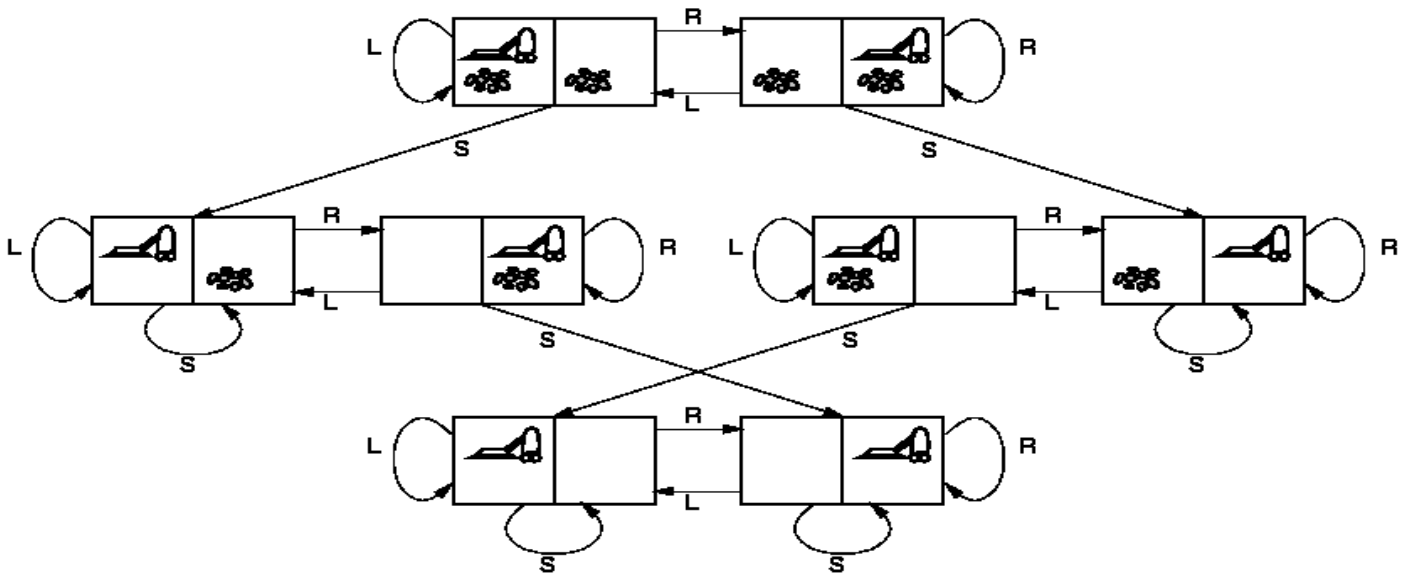
- Custo total da busca =
  - custo de busca (tempo e memória, i.e. custo computacional) -> busca da solução
  - + custo do caminho -> execução da solução
- Espaço de estados grande
  - compromisso (conflito) entre determinar
    - a melhor solução em termos de custo do caminho (é uma boa solução?) e
    - a melhor solução em termos de custo computacional (é computacionalmente barata?)

Abstração

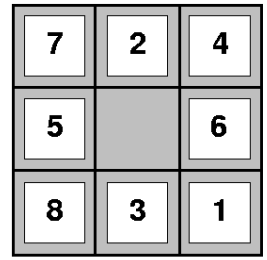
- Abstração *válida*: se qqr solução abstrata pode ser expandida em uma solução mais detalhada;
- Abstração *útil*: se a execução de cada uma das ações na solução é mais fácil que o problema original.
  - i.e., os passos da solução abstrata não exigem (muito) planejamento adicional

Miniproblemas: Aspirador de pó

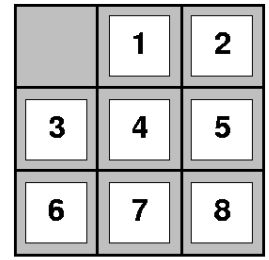
- **Estados:** duas posições para o agente
- **Estado inicial**: qqr
- **Função sucessor**: ações esq., dir., aspirar
- **Teste**: todos os quadrados limpos
- **Custo**: cada passo=1



Exemplos de Formulação de problema
Jogo de 8 números



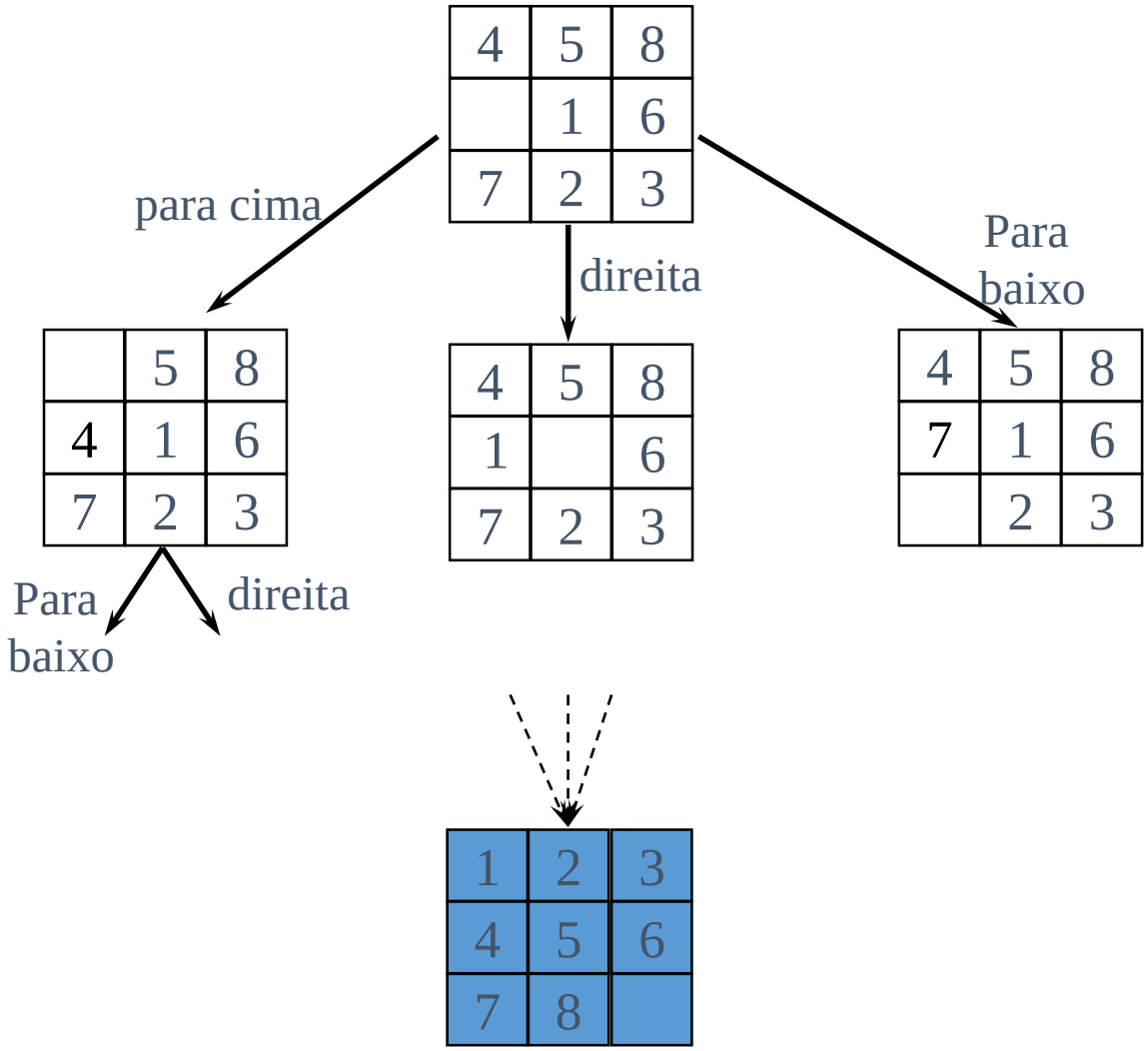
Start State



Goal State

- Espaço de estados = todas as possíveis configurações do tabuleiro
- Estado inicial = qualquer um dos estados possíveis
- Teste de término = tabuleiro ordenado, com branco na posição [3,3]
- Ações/operadores = mover peças numéricas para espaços livres (em branco) (esquerda, direita, para cima e para baixo)
- Custo do caminho = número de passos da solução
- Custo de busca = depende do computador e da estratégia de busca utilizada
  - Próximas aulas

Árvore de busca para o Jogo dos 8 números



Exemplos de formulação de problema

- Dirigir de Recife (PE) a Juazeiro do Norte (CE)
  - Espaço de estados = todas as cidades do mapa alcançáveis a partir do estado inicial
  - Estado inicial = estar em Recife
  - Teste de término (já atingimos o objetivo?) = estar em Juazeiro do Norte
  - Ações/operadores = dirigir de uma cidade para outra (se houver estrada entre elas!)
  - Função Custo do caminho = número de cidades visitadas, distância percorrida, tempo de viagem, grau de divertimento, etc

Custo do caminho diferente => Solução diferente

- Função de *custo de caminho*

  (1) distância entre as cidades

  (2) tempo de viagem, etc.

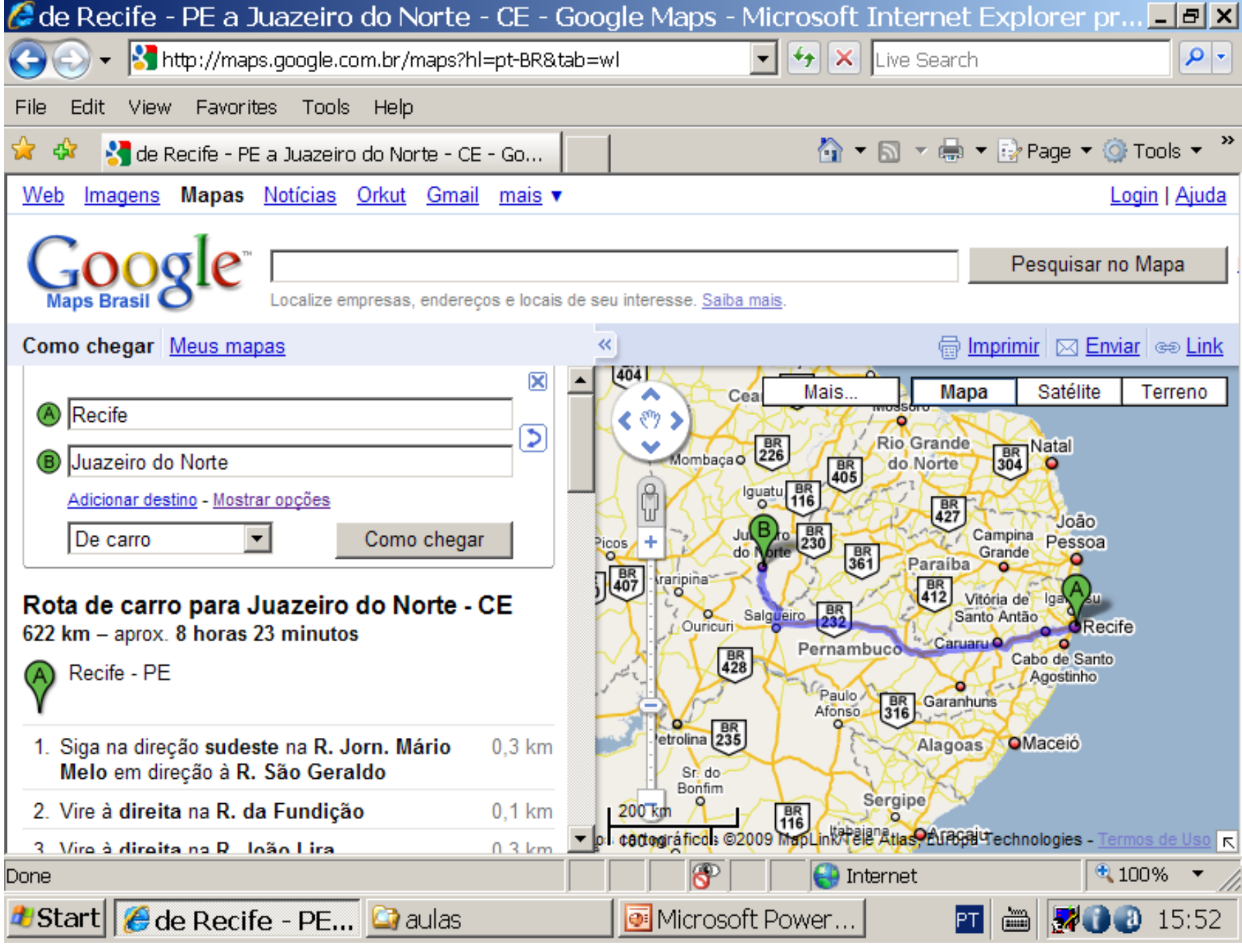
- Solução mais barata:

  (1) Camaragibe, Carpina, Patos, Milagres,...

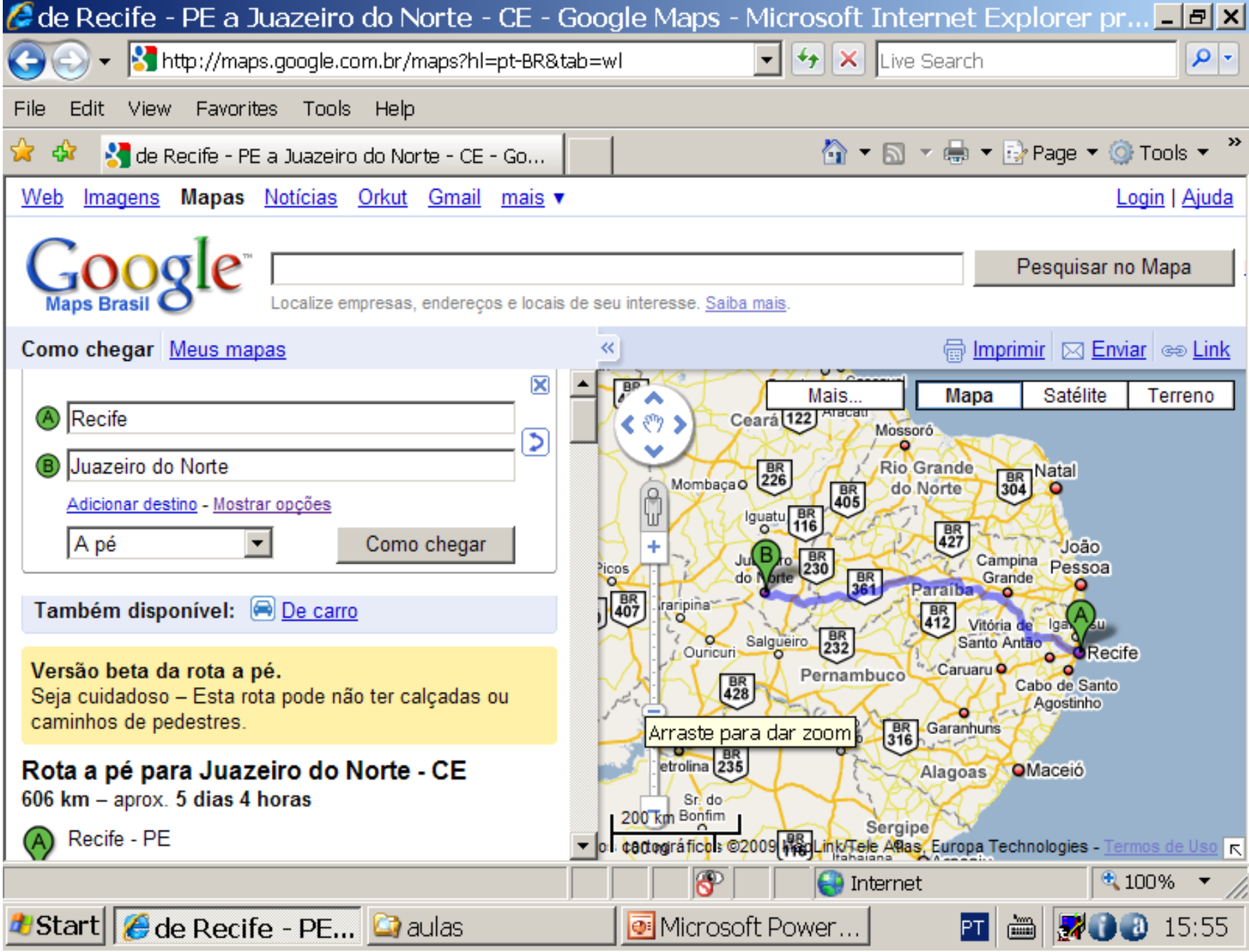
  (2) Moreno, Vitória de S. Antão, Caruaru, Salgueiro,...

  apesar de mais longa, pega estradas melhores e evita as cidades.

Recife – Juazeiro do Norte

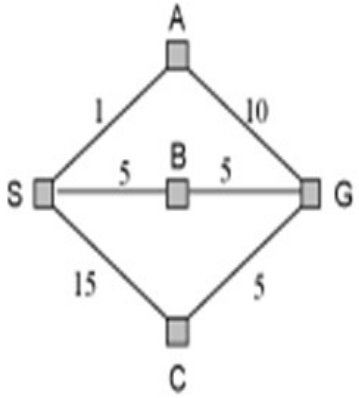


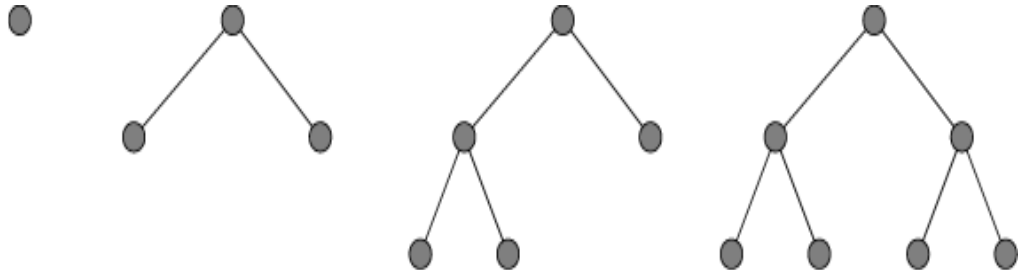
Recife – Juazeiro do Norte



Considere o seguinte problema:

- Encontrar o caminho de S para G de menor custo (menor soma das arestas)



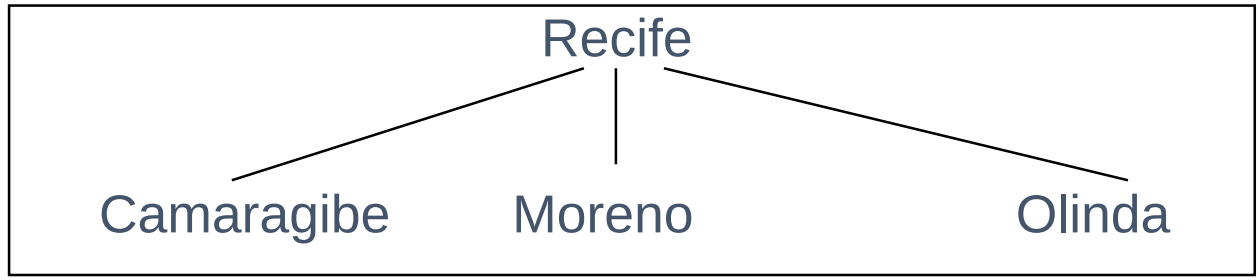


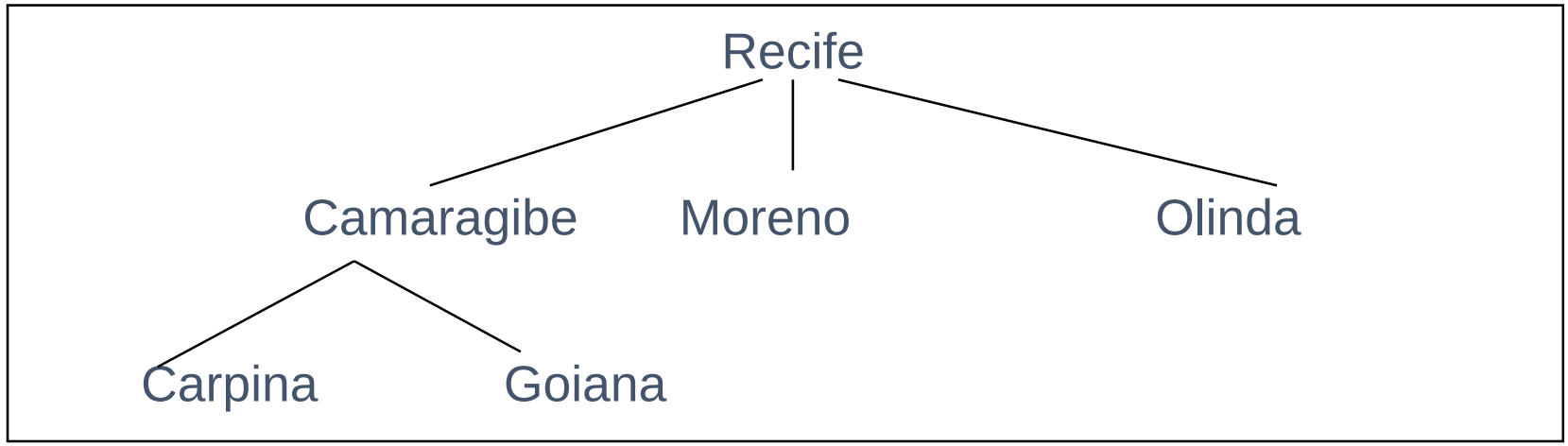
- Armazenar uma lista nos já visitados, ou sendo visitados atualmente: {A, B, C,...}
- Percorrer os nós da arvore de busca seguindo uma determinada ordem de expansão em uma determinada direção de expansão.

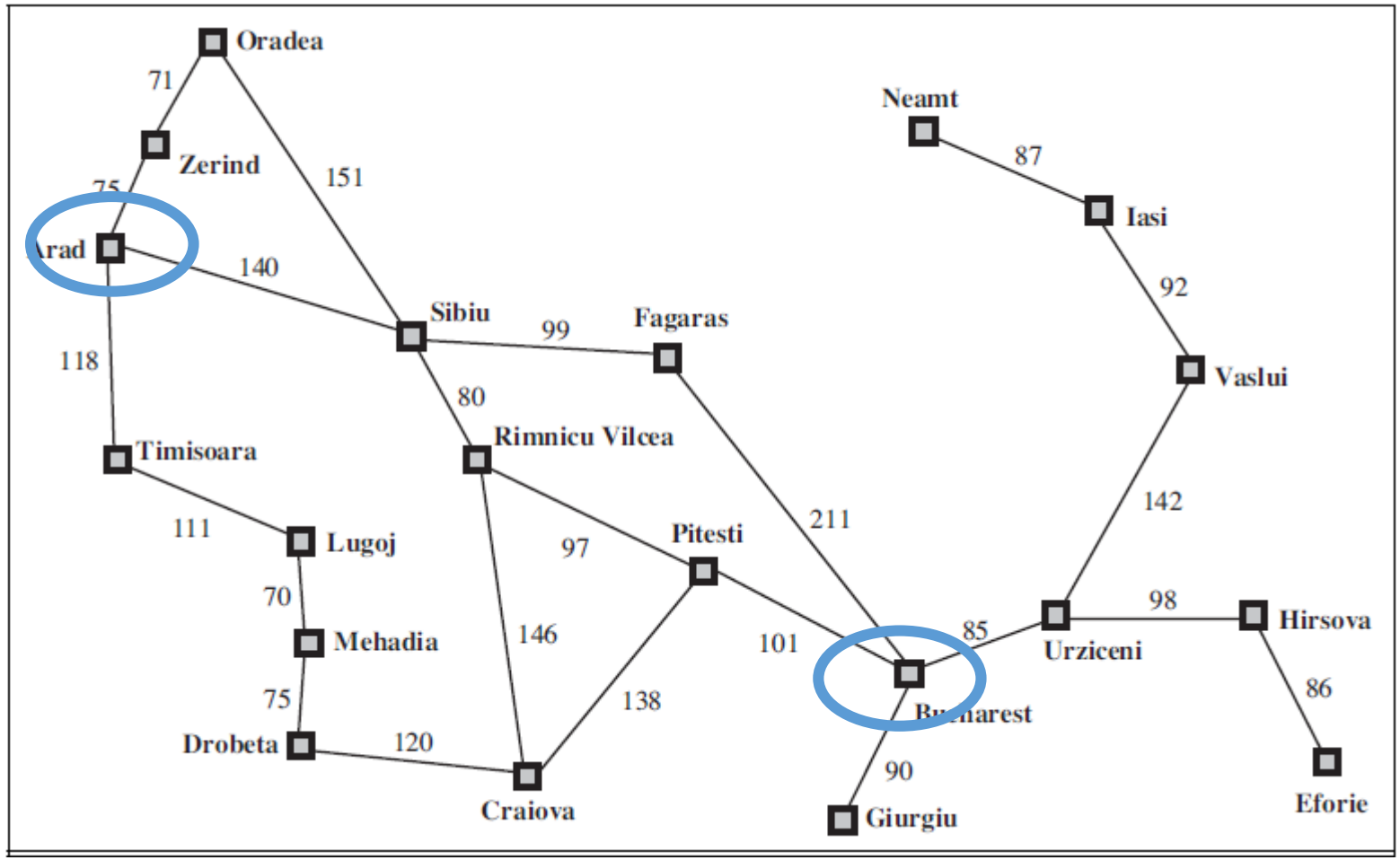
Exemplo: viajar de Recife a Juazeiro

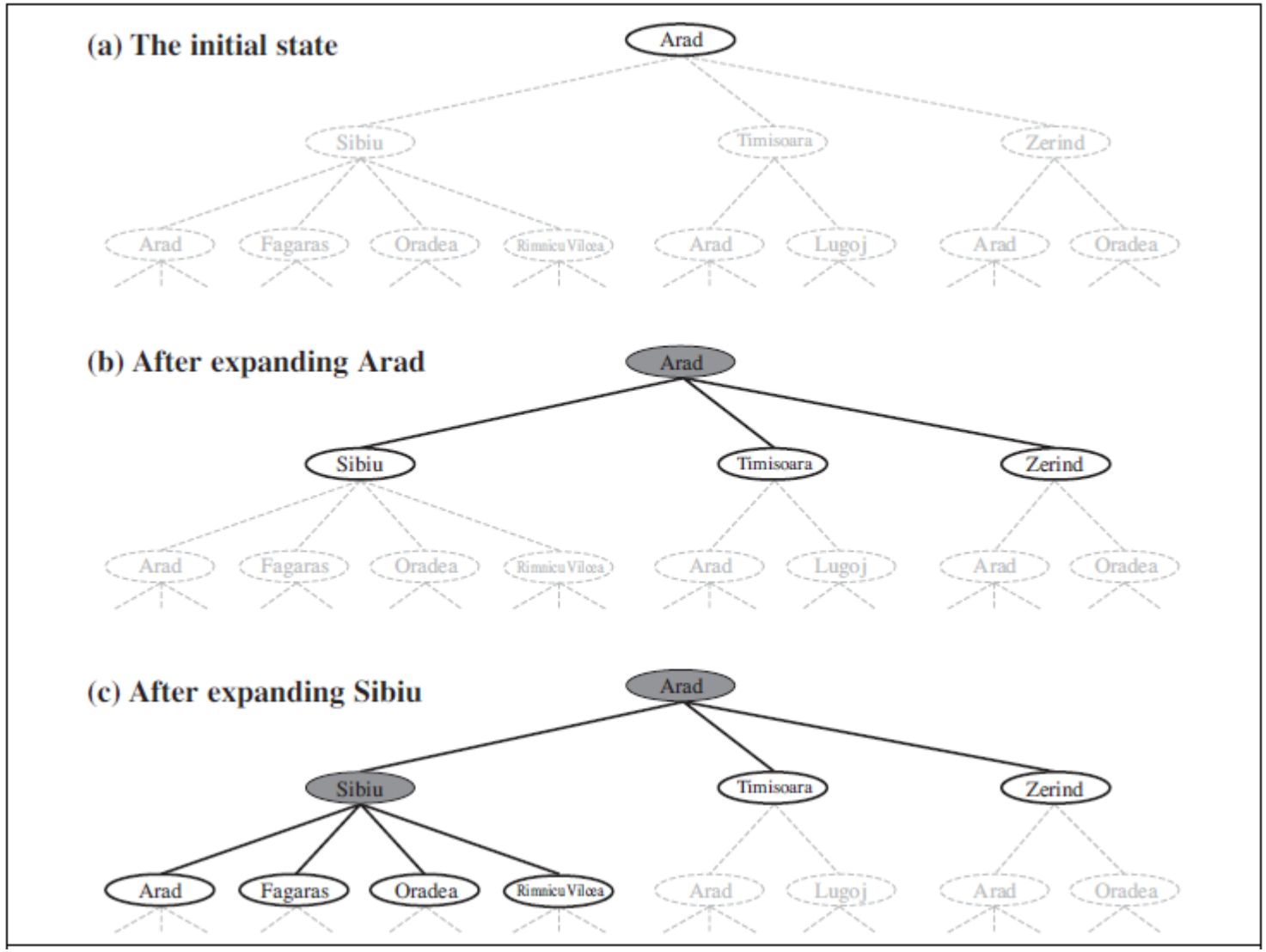
Árvore de Busca

Estado inicial => Recife

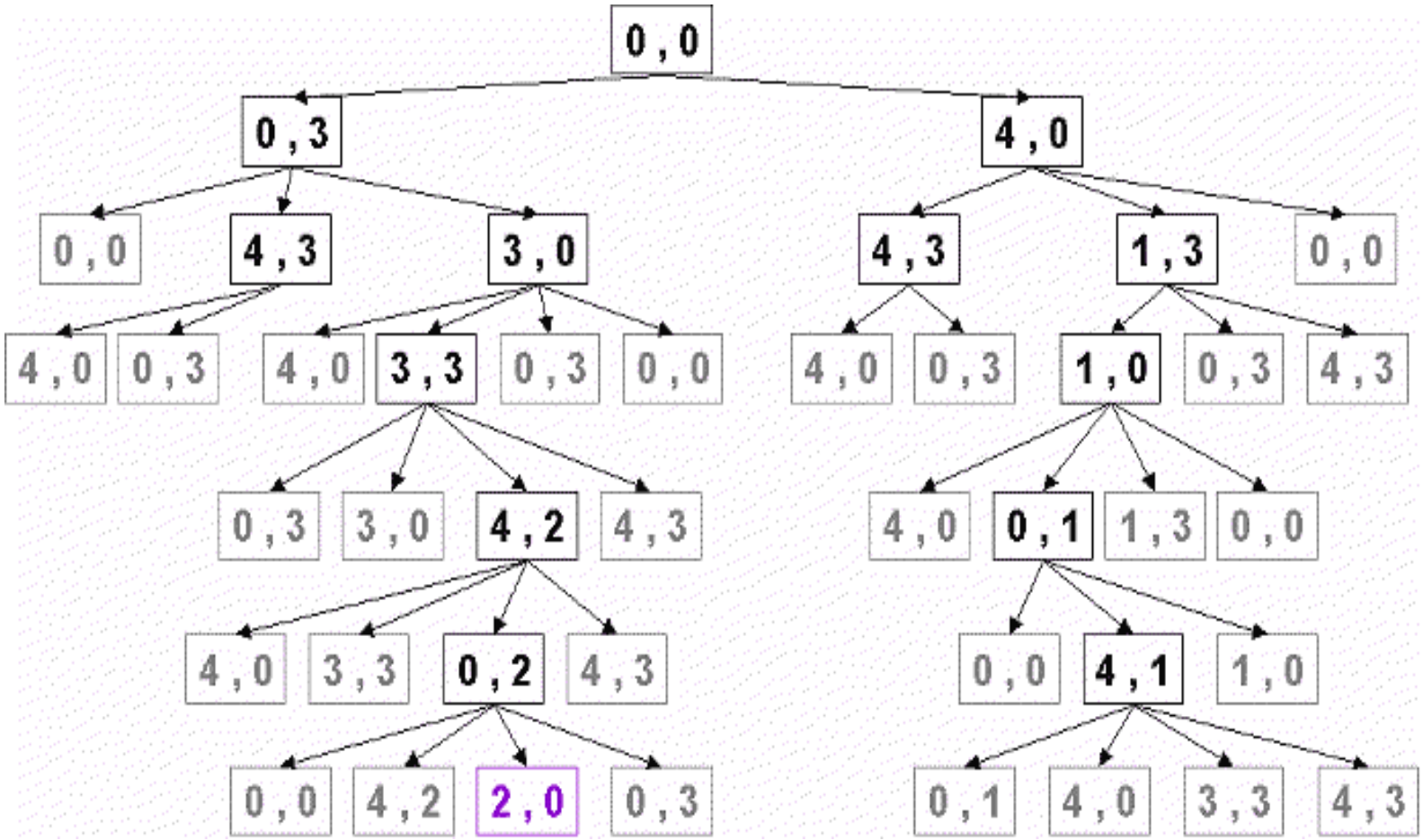





Oradea
71
Neamt
Zerind
87
151
Iasi
Arad
140
92
Sibiu
99
Fagaras
118
Vaslui
80
Rimnicu Vilcea
Timisoara
142
111
211
Lugoj
97
Pitesti
70
98
Hirsova
Mehadia
146
101
85
Urziceni
75
86
138
Drobeta
120
90
Craiova
Giurgiu
Eforie


(a) The initial state
Arad
Sibiu
Timisoara
Zerind
Arad
Fagaras
Oradea
Rimnicu Vilcea
Arad
Lugoj
Arad
Oradea
(b) After expanding Arad
Arad
Sibiu
Timisoara
Zerind
Arad
Fagaras
Oradea
Rimnicu Vilcea
Arad
Lugoj
Arad
Oradea
(c) After expanding Sibiu
Arad
Sibiu
Timisoara
Zerind
Arad
Fagaras
Oradea
Rimnicu Vilcea
Arad
Lugoj
Arad
Oradea

Exemplo dos Jarros

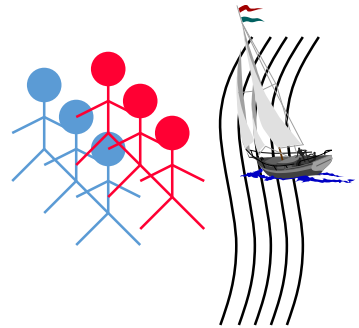


Aplicações de Busca: “Toy Problems”

- Jogo das 8 rainhas
- Jogo dos n números (*n-puzzle*)
- Criptoaritmética

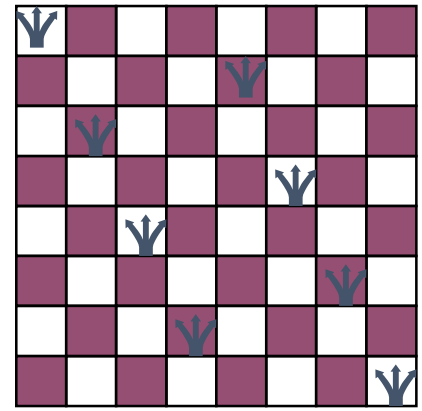
```
  send
+ more
--------
 money
```

- Palavras cruzadas
- Canibais e missionários



Importância da formulação
Ex.: Jogo das 8 Rainhas

- Objetivo: dispor 8 rainhas no tabuleiro sem possibilitar “ataques”
  - i.e., não pode haver mais de uma rainha em uma mesma linha, coluna ou diagonal



- Existem diferentes estados e operadores possíveis
  - essa escolha pode ter consequências boas ou nefastas na complexidade da busca ou no tamanho do espaço de estados

Formulações para 8 Rainhas

- Formulação A
  - estados: qualquer disposição com n ($n \leq 8$) rainhas
  - operadores: adicionar uma rainha a qualquer quadrado
  - 64x63x...x57 = $3x10^{14}$ possibilidades: vai até o fim para testar se dá certo
- Formulação B
  - estados: disposição com n ($n \leq 8$) rainhas sem ataque mútuo (teste gradual)
  - operadores: adicionar uma rainha na coluna vazia mais à direita em que não possa ser atacada
  - melhor (2057 possibilidades), mas pode não haver ação possível
- Formulação C
  - estados: disposição com 8 rainhas, uma em cada coluna
  - operadores: mover uma rainha atacada para outra casa na mesma coluna

Aplicações de Busca: Problemas Reais

- Cálculo de rotas
  - encontrar a melhor rota de um ponto a outro (aplicações: redes de computadores, planejamento militar, planejamento de viagens aéreas)
    - rotas em redes de computadores
    - sistemas de planejamento de viagens
    - planejamento de rotas de aviões
- Turismo
  - visitar cada ponto pelo menos uma vez
- Caixeiro viajante
  - visitar cada cidade exatamente uma vez
  - encontrar o caminho mais curto
- Alocação (Scheduling)
  - Salas de aula
  - Máquinas industriais (job shop)
- Projeto de VLSI
  - Cell layout
  - Channel routing

Aplicações de Busca: Problemas Reais

- Navegação de robôs:
  - generalização do problema da navegação
  - robôs movem-se em espaços contínuos, com um conjunto (infinito) de possíveis ações e estados
  - controlar os movimentos do robô no chão, e de seus braços e pernas requer espaço multi-dimensional
- Montagem de objetos complexos por robôs:
  - ordenar a montagem das diversas partes do objeto
- etc...

Solucionando o problema:
três passos fundamentais: formulação, busca e execução

- Formulação do problema e do objetivo (manual)
  - quais são os estados e as ações a considerar?
  - qual é (e como representar) o objetivo?
- Busca (processo automático)
  - processo que gera/analisa seqüências de ações para alcançar um objetivo
  - solução = caminho entre estado inicial e estado final.
- Execução (manual ou automática)

Busca em Espaço de Estados

- Depois de formular adequadamente o problema, a solução deve ser "buscada" automaticamente
  - Solução: caminho (seqüência de ações) que leva do estado inicial a um estado final (objetivo).
- Deve-se usar um método de busca para determinar a (melhor) solução para o problema
- Uma vez a busca terminada com sucesso, é só executar a solução
  - De forma manual ou automática (ex., um robô)

Busca em Espaço de Estados
Algoritmo de geração e teste

- **Fronteira** do espaço de estados
  - Lista contendo os nós (estados) a serem expandidos
  - Inicialmente, a fronteira contém apenas o estado inicial do problema
- **Algoritmo:**

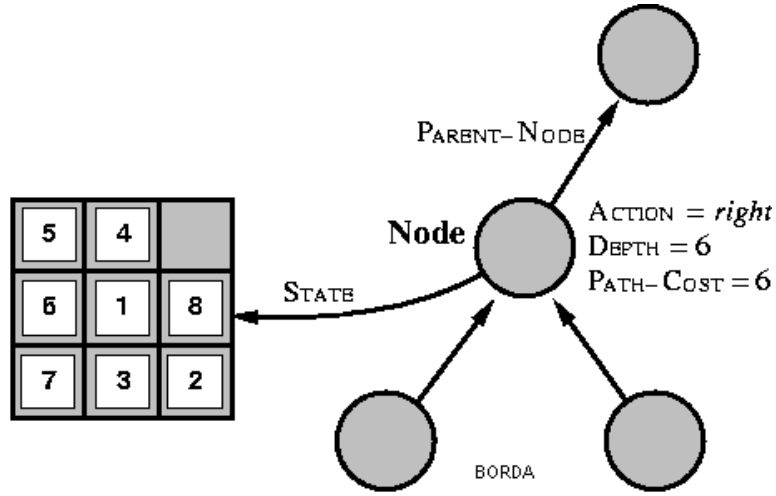
1. Selecionar o primeiro nó (estado) da fronteira do espaço de estados;
   - se a fronteira está vazia, o algoritmo termina com falha.
2. Testar se o nó selecionado é um estado final (objetivo):
   - se "sim", então retornar nó - a busca termina com sucesso.
3. Gerar um novo conjunto de estados aplicando ações ao estado selecionado;
4. Inserir os nós gerados na fronteira, de acordo com a estratégia de busca usada, e voltar para o passo (1).

Busca em Espaço de Estados
Implementação do algoritmo

- Os nós da fronteira devem guardar mais informação do que apenas o estado:

  → Na verdade nós são uma estrutura de dados com 5 componentes:

  1. o estado (configuração) correspondente ao nó atual
  2. o seu nó pai – ou o caminho inteiro para não precisar de operações extras
  3. a ação aplicada ao pai para gerar o nó – verifica de onde veio para evitar loops
  4. o custo do nó desde a raiz (*g(n)*)
  5. a profundidade do nó – se guardar o caminho não precisa!



Busca em Espaço de Estados
Implementação do algoritmo

Função-Insere: controla a ordem de inserção de nós na fronteira do espaço de estados.

```
função Busca-Genérica (problema formulado, Função-Insere)
    retorna uma solução ou falha
        fronteira ← Estado-Inicial (problema)
        loop do
            se fronteira está vazia então retorna falha
            nó ← Remove-Primeiro (fronteira)
            se Teste-Término (problema, nó) tiver sucesso
                então retorna nó
            fronteira ← Função-Insere (fronteira, Ações (nó) )
        end
```

Métodos de busca

- Busca exaustiva (cega)
  - Não sabe qual o melhor nó da fronteira a ser expandido
    - i.e., menor custo de caminho desse nó até um nó final (objetivo).
  - Estratégias de Busca (ordem de expansão dos nós):
    - caminhamento em largura
    - caminhamento em profundidade

- Busca heurística (informada)
  - Estima qual o melhor nó da fronteira a ser expandido com base em funções heurísticas => conhecimento
  - Estratégia de busca: *best-first search* (melhor escolha)

Critérios de Avaliação das Estratégias de Busca

- Completude:
  - a estratégia sempre encontra uma solução quando existe alguma?
- Qualidade (“otimalidade” - *optimality*):
  - a estratégia encontra a melhor solução quando existem diferentes soluções?
  - i.e., solução de menor custo de caminho
- Custo do tempo:
  - quanto tempo gasta para encontrar a 1ª solução?
- Custo de memória:
  - quanta memória é necessária para realizar a busca?

Próxima aula

- Busca Cega e Busca Heurística
- Lembre de imprimir os slides da próxima aula antes dela!