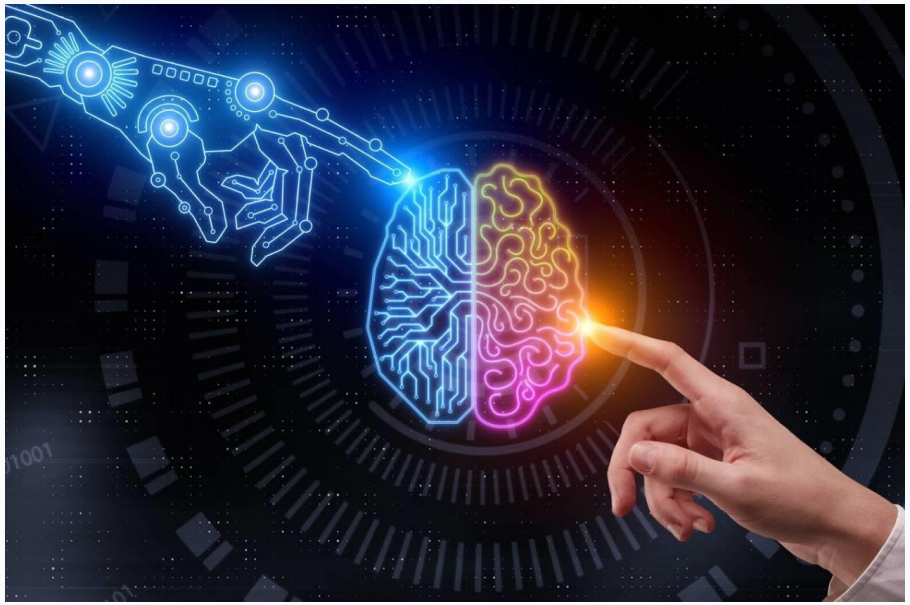


Inteligência Artificial

Profº - Dr. Thales Levi Azevedo Valente

thales.l.a.valente@gmail.com.br

Governo Federal

Grupo da turma 2024.2

https://chat.whatsapp.com/JFB6CgOI7lMCoYmoIKEK62

Sejam Bem-vindos !

Os celulares devem ficar no silencioso ou desligados

Pode ser utilizado apenas em caso de emergência

Boa tarde/noite, por favor e com licença DEVEM ser usados

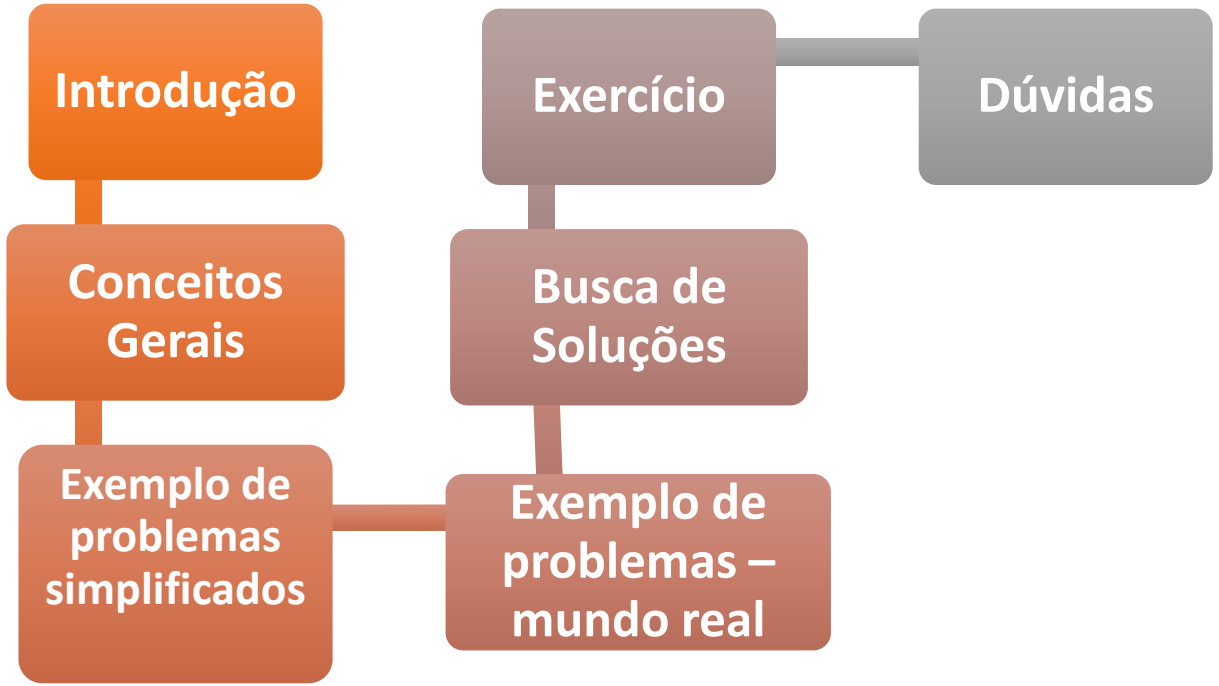
Educação é essencial

Objetivos de hoje

Fornecer uma visão abrangente sobre as metodologias de busca em Inteligência Artificial;

Ao final da aula, os alunos serão capazes de entender os principais conceitos conceitos introdutórios de algoritmos de busca e algumas aplicações tanto em ambientes simplificados quanto no mundo real.

Roteiro: Metodologias de busca



Introdução

> *"Research is the process of going up alleys to see if they are blind."*
> **(Marston Bates)**

MARSTON BATES
Professor of Zoology.
Link:
http://www.lib.umich.edu/faculty-history/faculty/marston-bates

Introdução

- **Agentes reativos (mais simples):** baseiam suas ações em uma mapeamento direto de estados em ações.
  - ✓ Não podem operar bem em ambientes com mapeamento grande demais para armazenar e levaria muito tempo para se aprender.
  - ✓ Não funcionam em ambientes para quais o número de regras condição-ação é grande demais para armazenar
- **Agentes baseados em objetivos:** consideram ações futuras e o quanto seus resultados são desejáveis.
  - ✓ Agente de resolução de problemas (*tipo de agente baseado em objetivo*).

Introdução

- Resolução por busca

- Um agente com várias opções imediatas pode decidir o que fazer comparando diferentes sequências de ações possíveis
- Esse processo de procurar pela melhor sequência é chamado de BUSCA
- O processo executado pelo agente segue a ordem
  - ✓ **Formular objetivo → buscar → executar**

Introdução

- Resolução por busca

- Definição precisa dos problemas
- Definição precisa das soluções
- Algoritmos de buscas
  - ✓ **Sem informação:** não fornece informações sobre o problema exceto sua definição
  - ✓ **Informada:** fornece orientação sobre onde procurar soluções

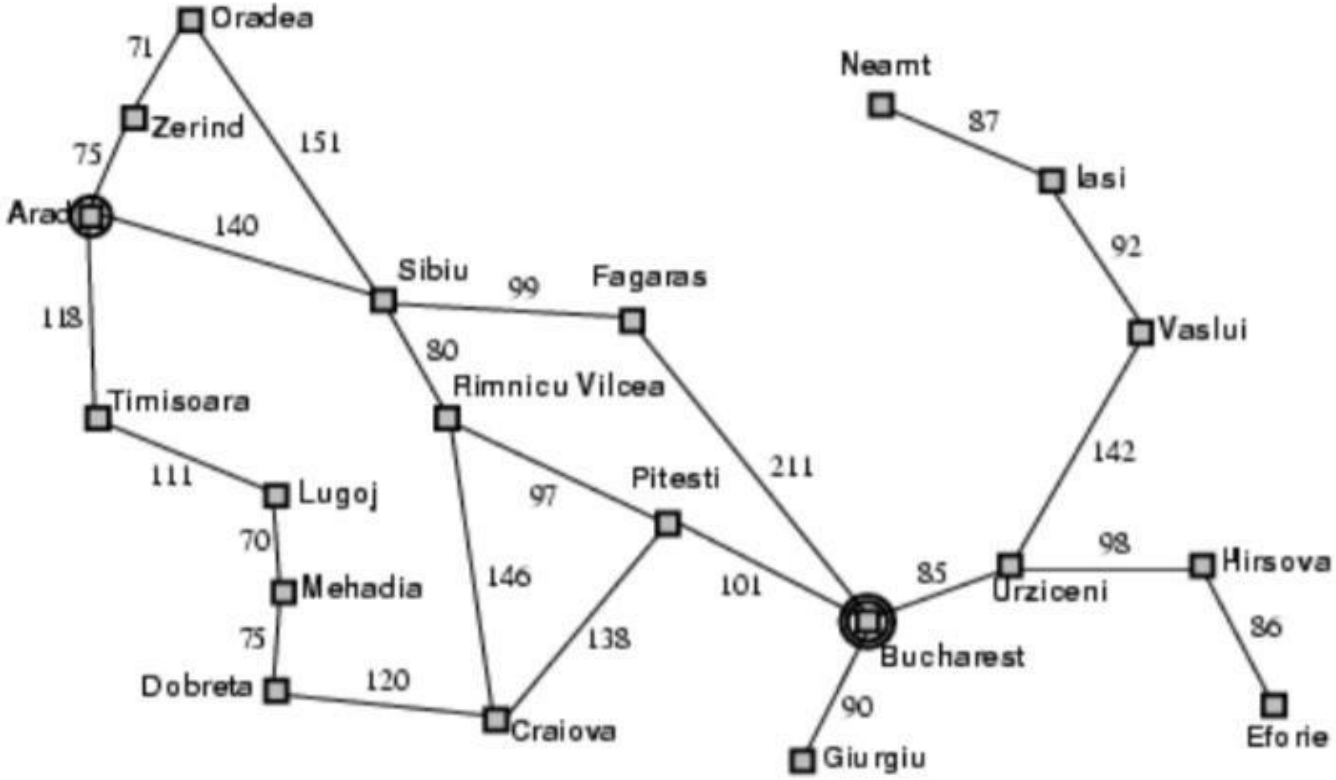
Conceitos Gerais

- Problemas e soluções definidos

- Um **problema** pode ser definido formalmente por cinco componentes:
  - ✓ **Estado inicial:** em que o agente começa.
  - ✓ **Ações:** descrição das ações possíveis que estão disponíveis para o agente.
  - ✓ **Modelo de transição:** descrição do que cada ação faz.
  - ✓ **Teste de objetivo:** determina se um estado é um estado objetivo.
  - ✓ **Uma função de custo de caminhos:** atribui um custo numérico a cada caminho.

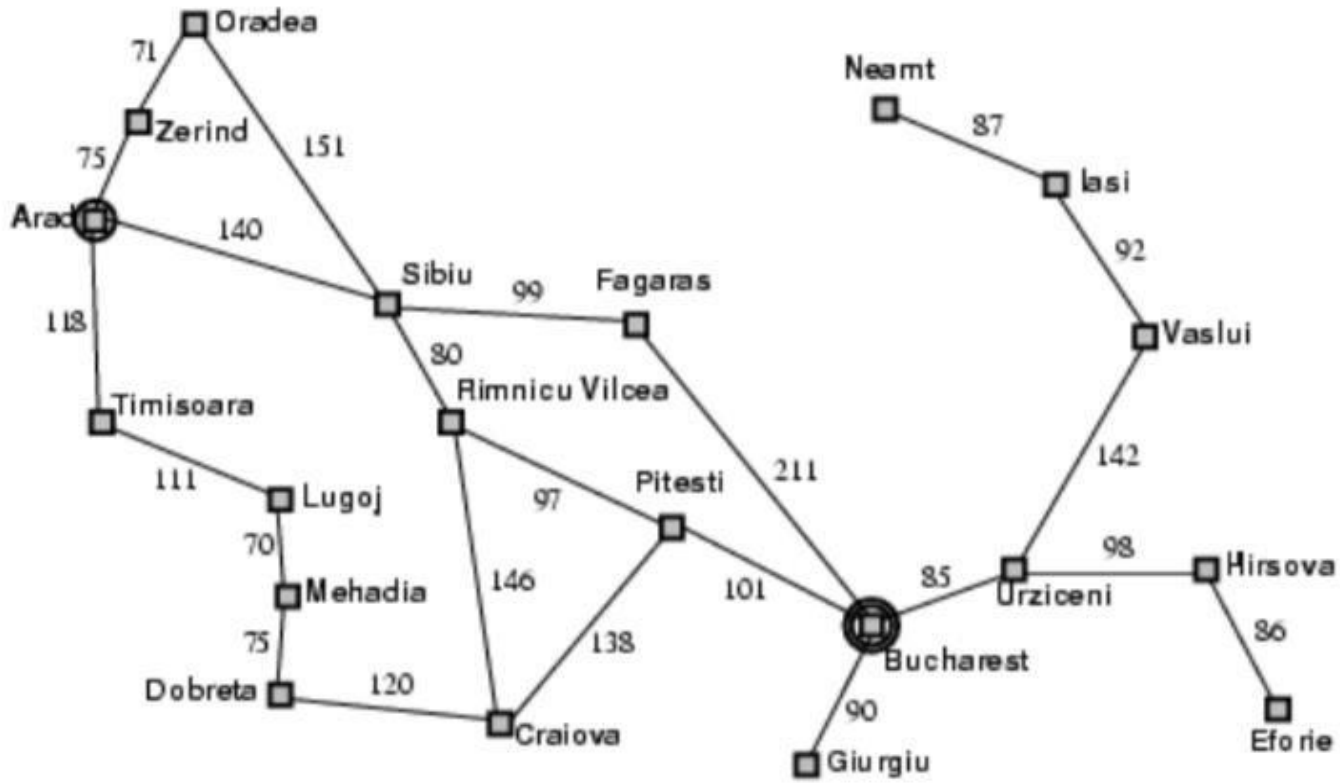
Exemplo: Romênia

- De férias na Romênia; atualmente em Arad
- Chegar em Bucareste
- Formular objetivo
  - ✓ **Estar em Bucareste**



Exemplo: Romênia

- Formular objetivo
  - ✓ **Estar em Bucareste**
- Formular problema
  - ✓ **Estados: cidades**
  - ✓ **Ações: dirigir entre as cidades**
- Encontrar solução
  - ✓ **– sequência de cidades**
  - ✓ **ex.: Arad, Sibiu, Fagaras, Bucareste**
  - ✓ **ex.: Arad, Sibiu, R. Vilcea, Pitesti, Bucarest**



Conceitos Gerais

- Formulação de problemas

- Um problema é definido por 4 itens:
  - ✓ **1. Estado inicial ex.: “em Arad“**
  - ✓ **2. Ações ou função sucessora S(x) = conjunto de pares estado-ação**
    - ✓ **– ex.: S(Arad) = {<Arad → Zerind, Zerind>, … }**
  - ✓ **3. Teste de objetivo, pode ser:**
    - ✓ **– explícito, ex.: x = “em Bucharest"**
    - ✓ **– implícito, ex.: Cheque-mate(x)**
  - ✓ **4. Custo de caminho (aditivo):**
    - ✓ **– ex.: soma das distâncias, número de ações executadas, etc.**
    - ✓ **– c (x, a, y) é o custo do passo, que deve ser sempre ≥ 0**

Conceitos Gerais

- Formulação de problemas

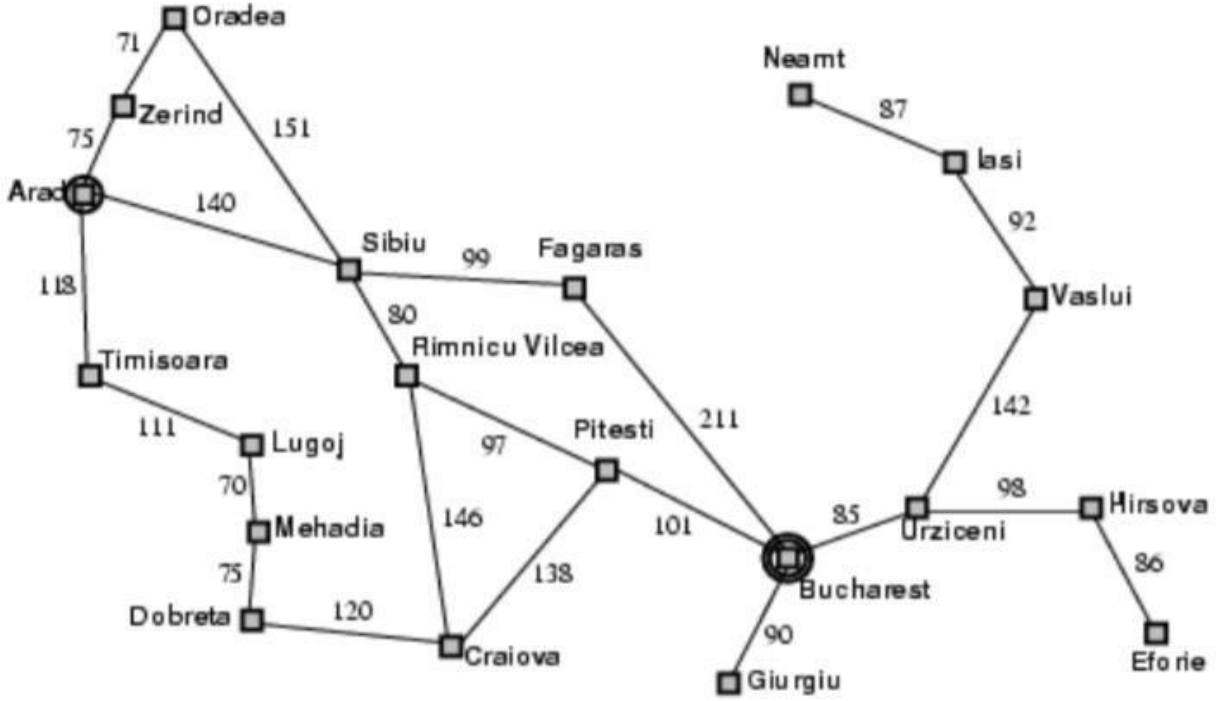
- Um problema é definido por 4 itens:
  - ✓ **1. Estado inicial ex.: "em Arad"**
  - ✓ **2. Ações ou função sucessora S(x) = conjunto de pares estado-ação**
    - ✓ **– ex.: S(Arad) = {<Arad → Zerind, Zerind>, … }**
  - ✓ **3. Teste de objetivo, pode ser:**
    - ✓ **– explícito, ex.: x = "em Bucharest"**
    - ✓ **– implícito, ex.: Cheque-mate(x)**
  - ✓ **4. Custo de caminho (aditivo):**
    - ✓ **– ex.: soma das distâncias, número de ações executadas, etc.**
    - ✓ **– c (x, a, y) é o custo do passo, que deve ser sempre ≥ 0**

- Uma solução é uma sequência de ações que levam do estado inicial para o estado objetivo.
- Uma solução ótima é uma solução com o menor custo de caminho.

Conceitos Gerais

- Formulação de problemas

- Uma solução é uma sequência de ações que levam do estado inicial para o estado objetivo.
- Uma solução ótima é uma solução com o menor custo de caminho.



Agente Simples para Resolução de Problema

```
function FUNCAO_DO_AGENTE (percepções) return ação
    static seq          //uma sequencia de ações, inicialmente vazia
           estado       //descrição do estado atual
           objetivo     //o objetivo
           problema     //a formulação do problema
    estado = ATUALIZAR_ESTADO (estado, percepções)
    if (seq == null) do
           objetivo = FORMULAR_OBJETIVO(estado)
           problema = FORMULAR_PROBLEMA(estado, objetivo)
           seq = BUSCA(problema)
    ação = FIRST(seq)
    seq = RESTO(seq)
    return ação
```

Conceitos Gerais

- Espaço de estados

- **O conjunto de todos os estados acessíveis a partir de um estado inicial é chamado de espaço de estados**
  - ✓ Os estados acessíveis são aqueles dados pela função sucessora

- **O espaço de estados pode ser interpretado como um grafo em que os nós são estados e as arestas são ações**

Conceitos Gerais

- Selecionando o espaço de estados

- **O mundo real é absurdamente complexo**
  - ✓ O espaço de estados é uma abstração
- **Estado (abstrato) = conjunto de estados reais**
- **Ação (abstrata) = combinação complexa de ações reais**
  - ✓ ex., "Arad → Zerind" representa um conjunto complexo de rotas, desvios, paradas,etc.
  - ✓ Qualquer estado real do conjunto "em Arad" deve levar a algum estado real "em Zerind".
- **Solução (abstrata) = conjunto de caminhos reais que são soluções no mundo real**
- **A abstração é útil se cada ação abstrata é mais fácil de executar que o problema original.**

Exemplo de problemas

- **Mundo simplificado:**
  - ✓ Destinado a ilustrar ou exercitar diversos métodos de resolução de problemas
  - ✓ Pode ter uma descrição concisa e exata
- **Mundo real:**
  - ✓ Não apresentam uma única descrição consensual
  - ✓ Dar uma ideia geral de suas formulações

Exemplo de problemas

- Problemas de mundo simplificado

- Exemplo: aspirador de pó
  - ✓ **Estados:** É determinado tanto pela posição do agente como da sujeira. O agente está em uma entre duas posições, cada uma das quais pode conter sujeira ou não (2×2^2).
  - ✓ **Estado inicial:** Qualquer estado pode ser designado como o estado inicial.
  - ✓ **Ações:** Cada estado tem apenas três ações (esquerda, direita e aspirar).

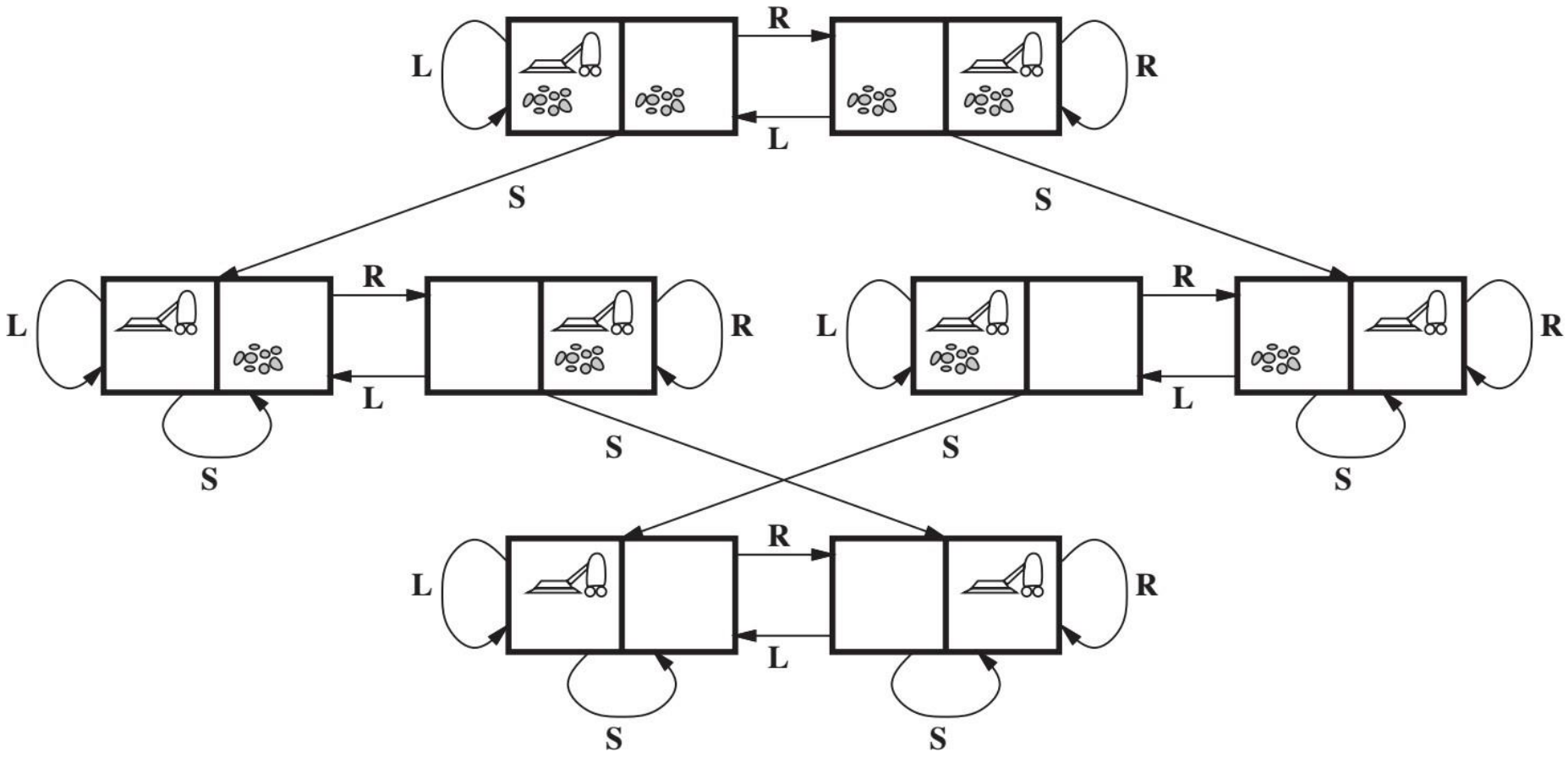
Exemplo de problemas

- Problemas de mundo simplificado

- Exemplo: aspirador de pó
  - ✓ **Modelo de transição:** As ações têm seus efeitos esperados.
  - ✓ **Teste de objetivo:** Verifica se todos os quadrados estão limpos.
  - ✓ **Custo de caminho:** Cada passo custa 1 e, assim, o custo do caminho é o número de passos do caminho.

Exemplo de problemas

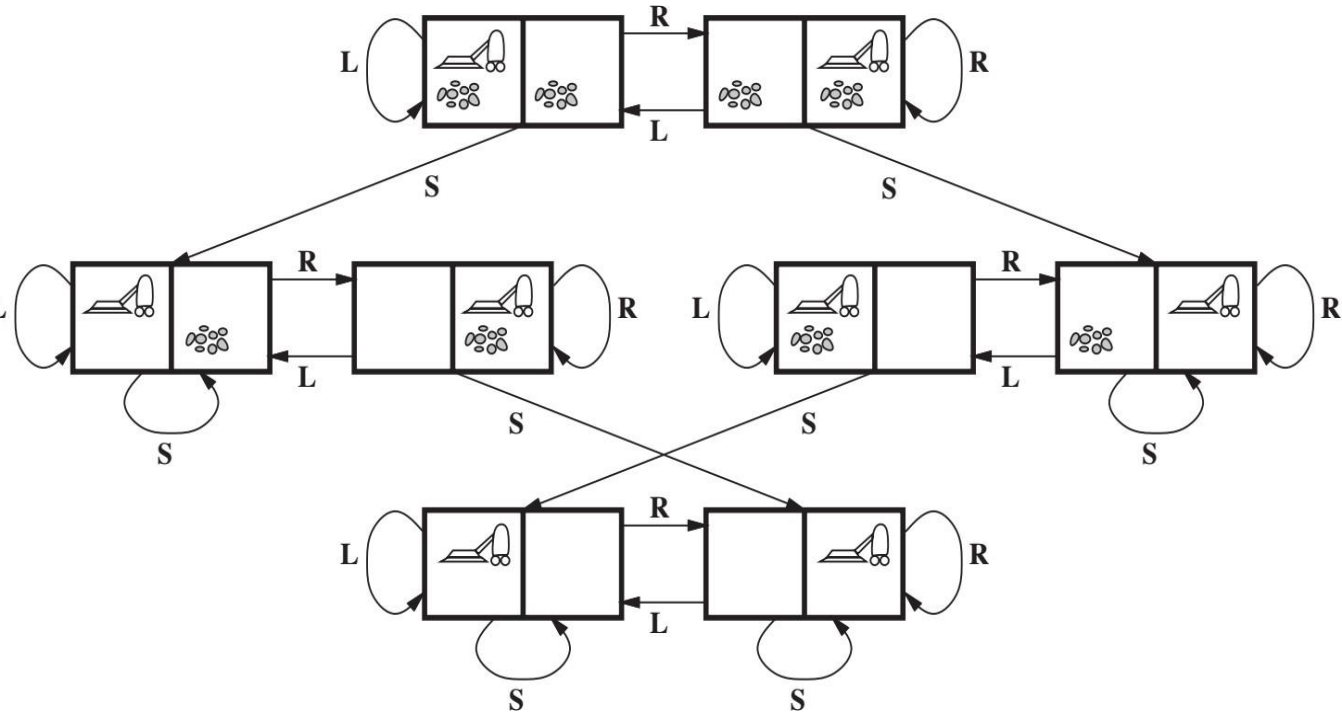
- Problemas de mundo simplificado



Exemplo de problemas

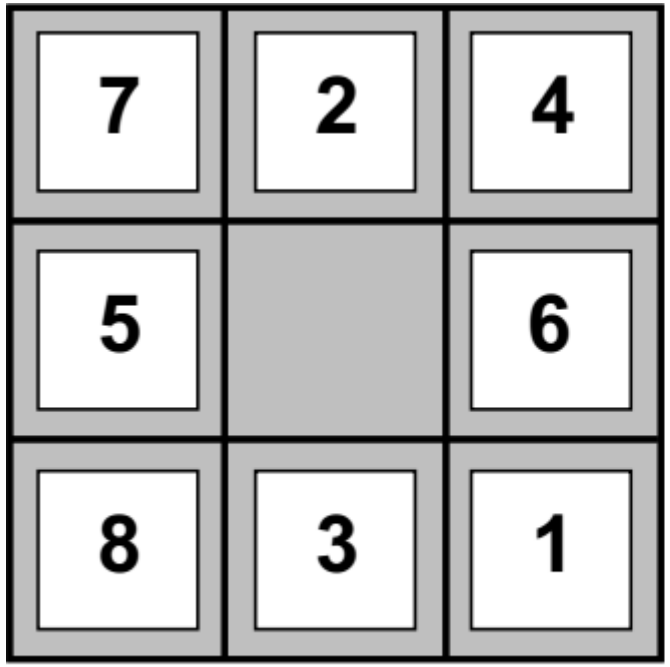
- Espaço de estados

- **Estados: Definidos pela posição do robô e sujeira (8 estados)**
- **Estado inicial: Qualquer um**
- **Função sucessor: pode-se executar qualquer uma das ações em cada estado (esquerda, direita, aspirar)**
- **Teste de objetivo: Verifica se todos os quadrados estão limpos**
- **Custo do caminho: Cada passo custa 1, e assim o custo do caminho é o número de passos do caminho**

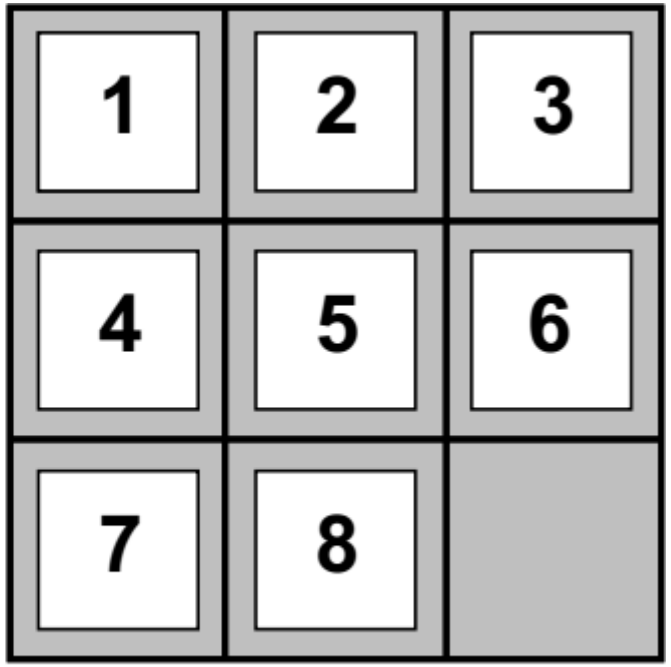


Exemplo de problemas

- Problemas de mundo simplificado



Start State

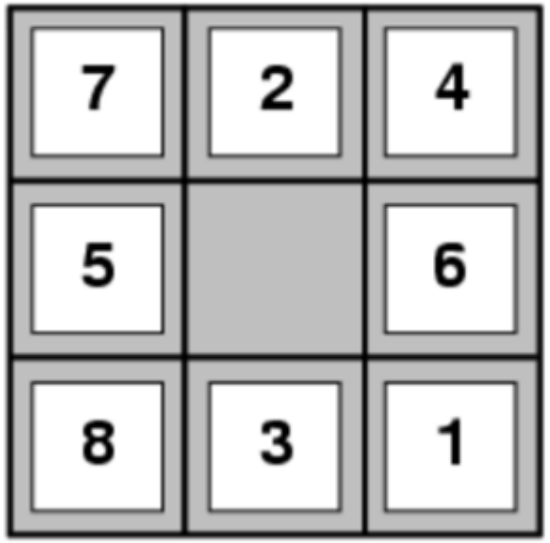


Goal State

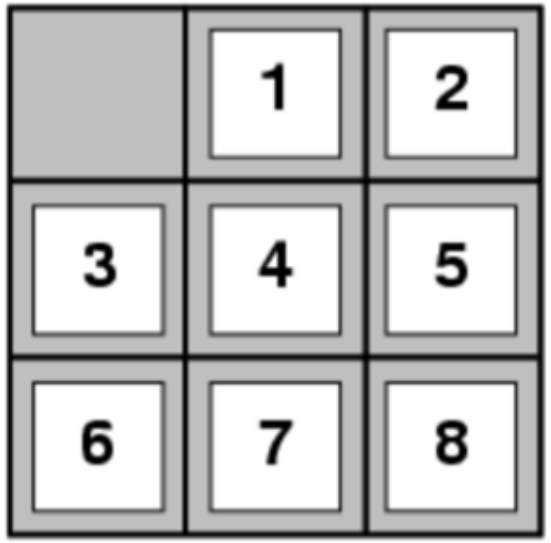
Exemplo de problemas

- Problemas de mundo simplificado

- **Estados: Especifica a posição de cada uma das peças e do espaço vazio**
- **Estado inicial: Qualquer um**
- **Função sucessor: gera os estados válidos que resultam da tentativa de executar as quatro ações (mover espaço vazio para esquerda, direita, acima ou abaixo)**
- **Teste de objetivo: Verifica se o estado corresponde à configuração objetivo.**
- **Custo do caminho: Cada passo custa 1, e assim o custo do caminho é o número de passos do caminho**



Start State



Goal State

Exemplo de problemas

- Problemas de mundo simplificado

- Exemplo: quebra cabeça de oito peças
  - ✓ **Estados:** Uma descrição de estado especifica a posição de cada uma das oito peças e do quadrado vazio em um dos nove quadrados.
  - ✓ **Estado inicial:** Qualquer estado pode ser designado como o estado inicial.
  - ✓ **Ações:** Movimentos do quadro vazio (esquerda, direita, para cima ou para baixo).

Exemplo de problemas

- Problemas de mundo simplificado

- Exemplo: quebra cabeça de oito peças
  - ✓ **Modelo de transição:** Dado um estado e ação, ele devolve o estado resultante.
  - ✓ **Teste de objetivo:** Verifica se o estado corresponde à configuração do estado desejado.
  - ✓ **Custo de caminho:** Cada passo custa 1 e, assim, o custo do caminho é o número de passos do caminho.

Exemplo de problemas

- Problemas do mundo real

- Exemplo: planejamento de viagem (viagens áreas)

Exemplo de problemas

- Problemas do mundo real

- Problema de roteamento
  - ✓ Encontrar a melhor rota de um ponto a outro (aplicações: redes de computadores, planejamento militar, planejamento de viagens aéreas)
- Problemas de tour
  - ✓ Visitar cada ponto pelo menos uma vez
- Caixeiro viajante
  - ✓ Visitar cada cidade exatamente uma vez
  - ✓ Encontrar o caminho mais curto

Exemplo de problemas

- Problemas do mundo real

- Layout de VLSI
  - ✓ Posicionamento de componentes e conexões em um chip
- Projeto de proteínas
  - ✓ Encontrar uma sequência de aminoácidos que serão incorporados em uma proteína tridimensional para curar alguma doença.
- Pesquisas na Web
  - ✓ É fácil pensar na Web como um grafo de nós conectados por link

Busca de soluções

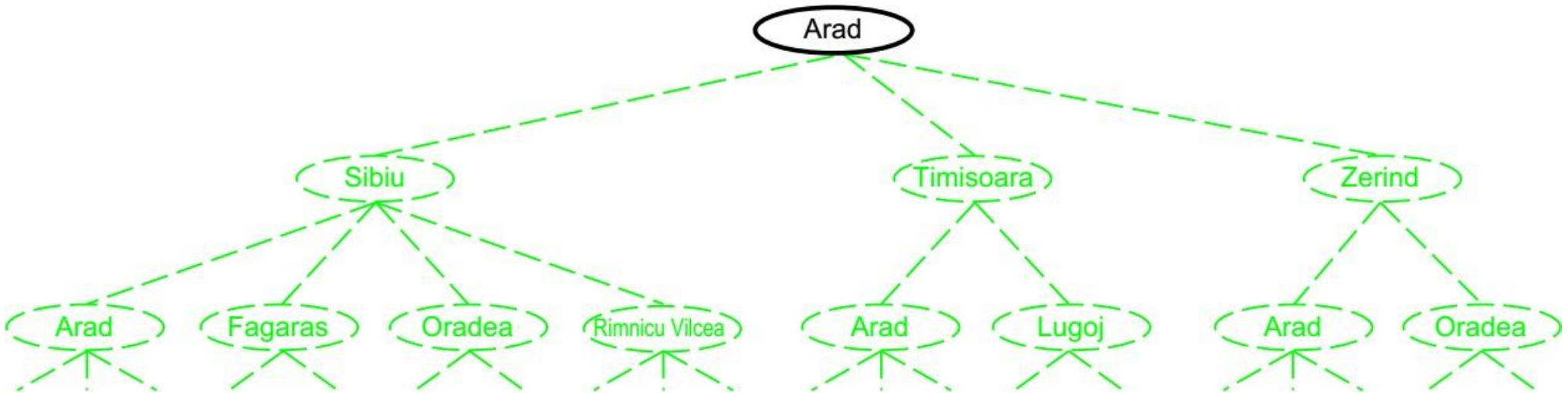
- Depois de formular alguns problemas, é necessário resolvê-los.
- Sequência de ações ou várias sequências de ações possíveis.
- Sequência de ações a partir de um ponto inicial.
  - ✓ **Árvores de busca:** estado inicial na raiz; os ramos são as ações, e os nós correspondem aos estados no espaço de estados do problema.

Importante: os algoritmos que se esquecem de sua história estão fadados a repeti-la.

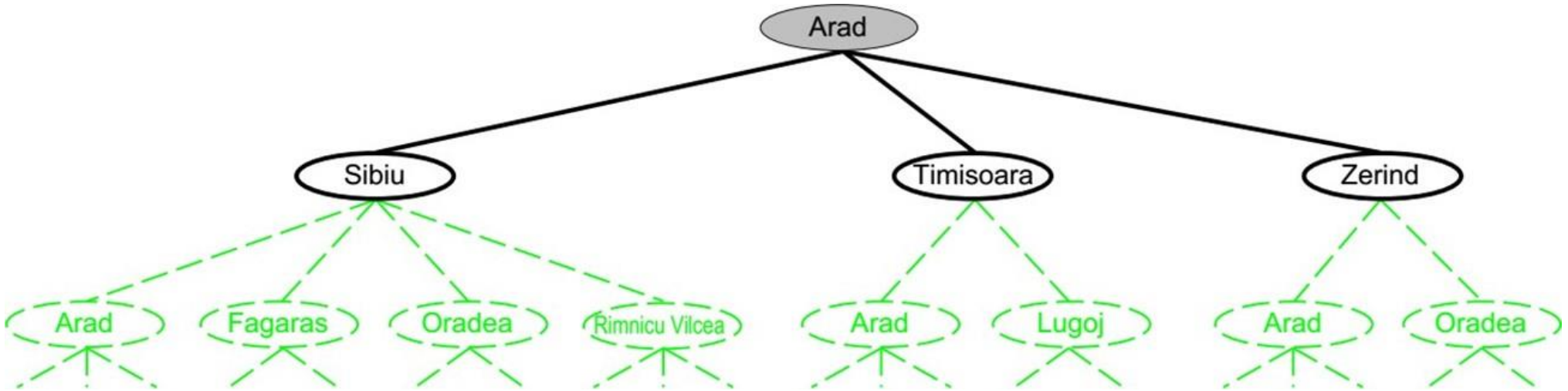
Busca de soluções

- Ideia: Percorrer o espaço de estados a partir de uma árvore de busca.
- Expandir o estado atual aplicando a função sucessor, gerando novos estados.
- Busca: seguir um caminho, deixando os outros para depois.
- A estratégia de busca determina qual caminho seguir

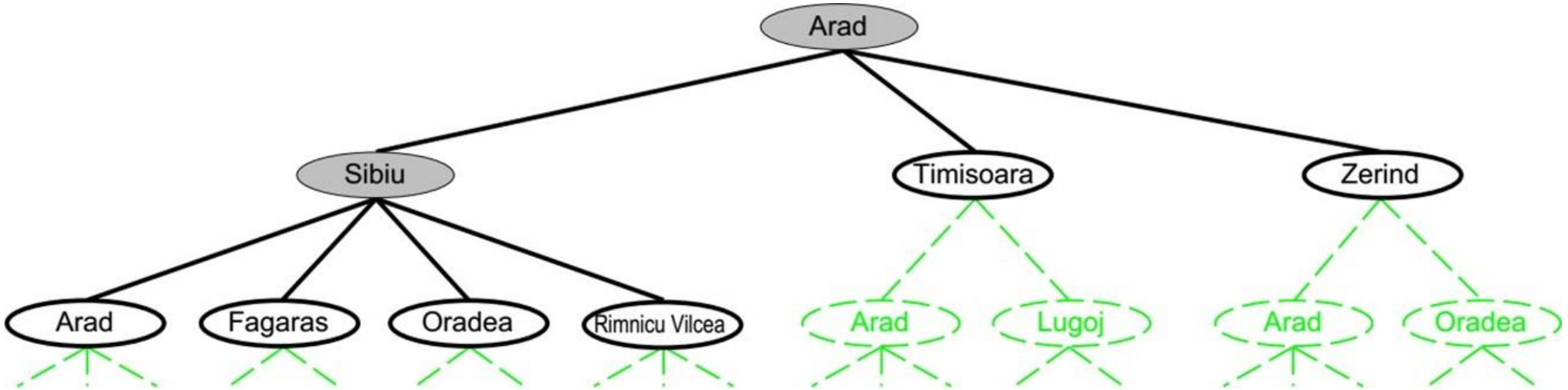
Em busca de soluções



Em busca de soluções



Em busca de soluções



Busca de soluções

- Infraestrutura para algoritmos de busca

- Estrutura de dados para manter o controle da árvore de busca que está sendo construída.
- Para cada nó *n* da árvore: estrutura de 4 componentes
  - ✓ *n*.**estado**: o estado no espaço que o nó corresponde
  - ✓ *n*.**pai**: o nó na árvore de busca que gerou esse nós
  - ✓ *n*.**ação**: a ação que foi aplicada ao pai para gerar o nó
  - ✓ *n*.**custo-do-caminho**: o custo do estado inicial até o nó

Busca de soluções

- Algoritmo genérico da árvore de busca

function ARVORE_DE_BUSCA (*problema*, *estratégia*) **return** *solução* ou *falha*

inicializa a árvore de busca usando o estado inicial do problema

while (*true*) **do**

if não existem candidatos para expandir **then**

return *falha*

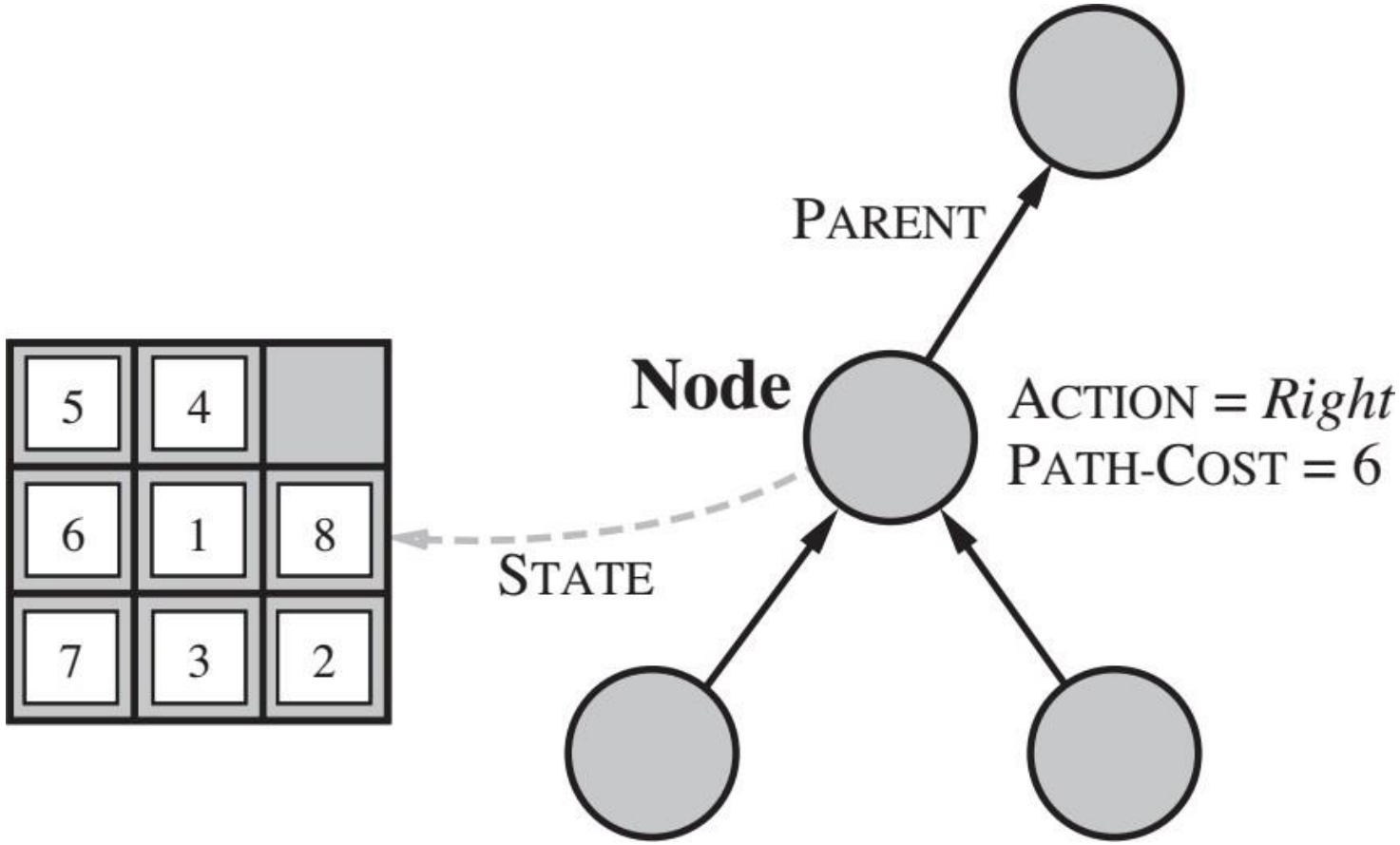
if o nó apresenta o estado objetivo **then**

return *solução*

expandir nó e adicionar o resultado na árvore de busca

Busca de soluções

- Infraestrutura para algoritmos de busca



Busca de soluções

- Medição de desempenho de resolução de problemas

- Uma estratégia de busca é definida pela escolha da ordem da expansão de nós
- Critérios de avaliação de desempenho (4 aspectos):
  - ✓ **Completeza:** o algoritmo oferece a garantia de encontrar uma solução quando ela existir?
  - ✓ **Otimização:** a estratégia encontra a solução ótima?
  - ✓ **Complexidade de tempo:** quanto tempo ele leva para encontrar uma solução?
  - ✓ **Complexidade de espaço:** quanta memória é necessária para executar a busca?

Exercícios

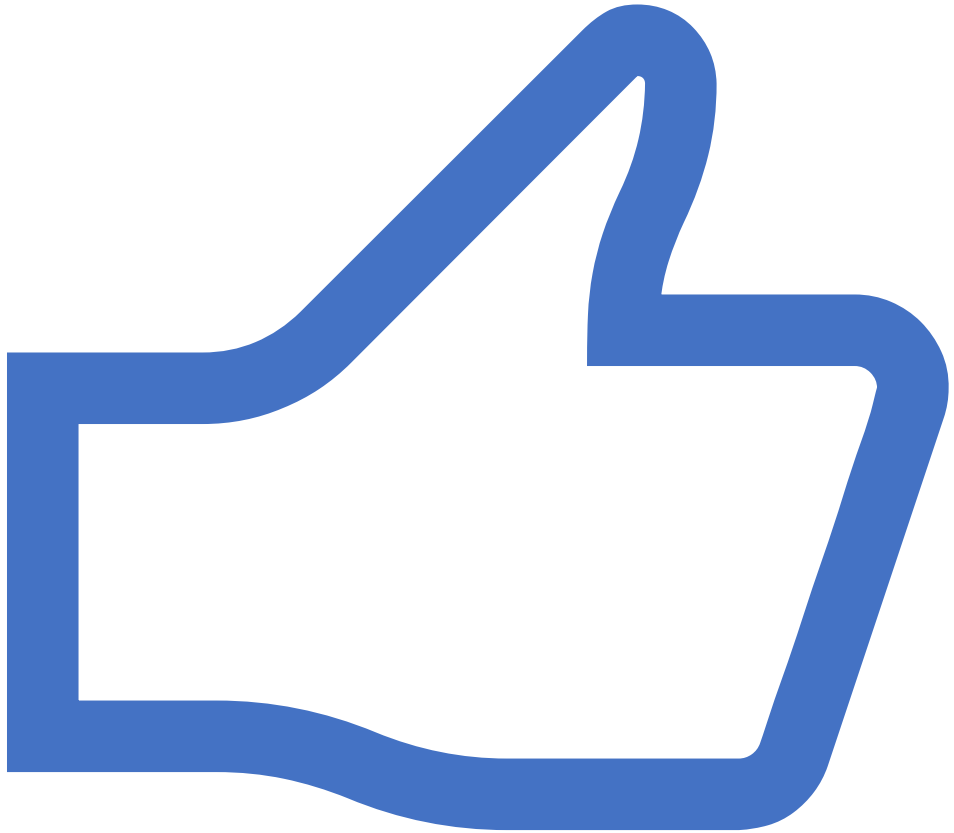
1) Explique por que a formulação do problema deve seguir a formulação do objetivo.
2) Formule dois tipos de problemas de mundo simplificado utilizando os critérios abordados em sala de aula.
3) Crie um algoritmo de busca para solucionar o problema das cidades da Romênia.

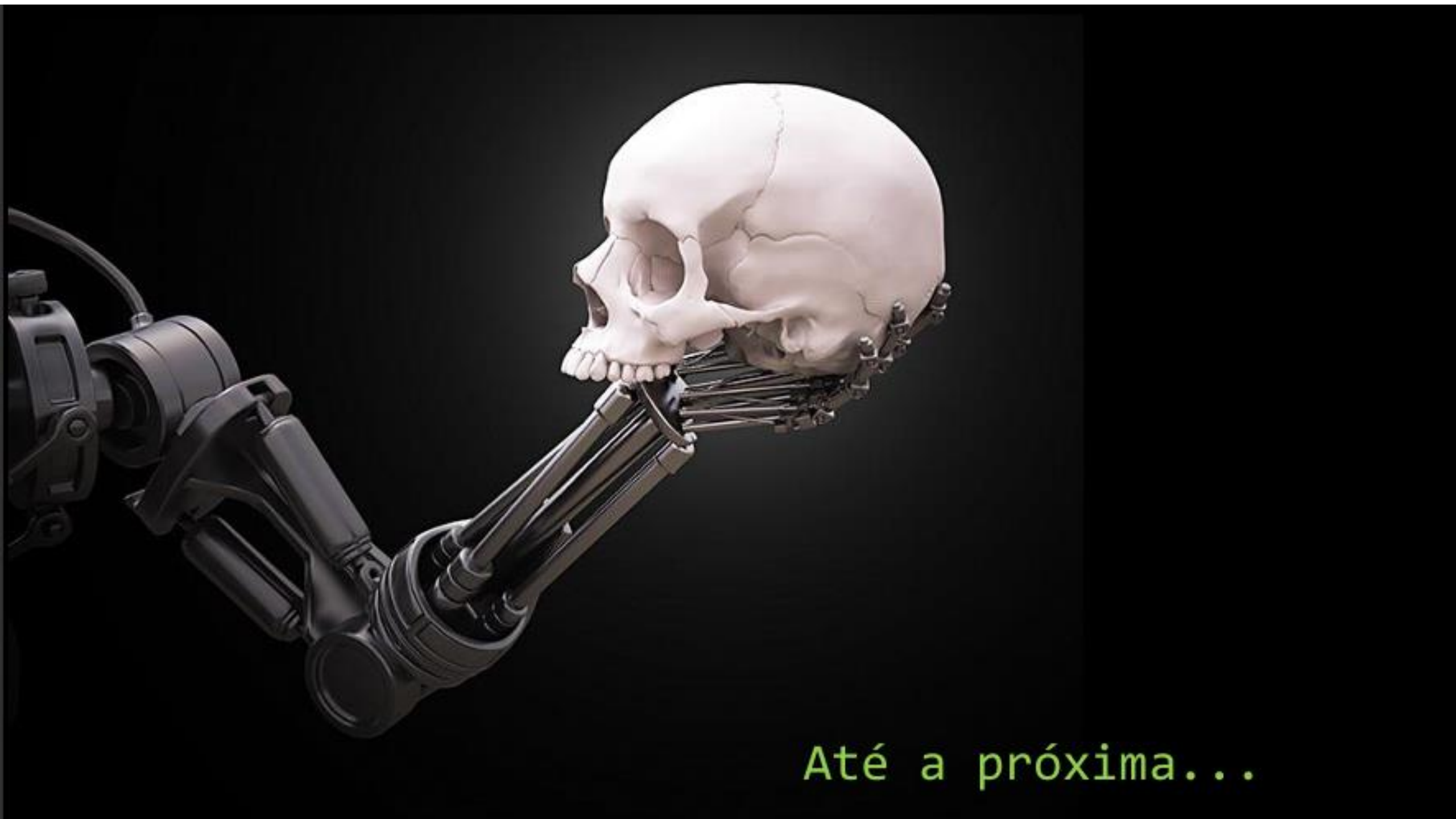
Dúvidas?

Obrigado!

Você disse que eu deveria passar mais tempo com as nossas crianças, então eu transformei a cara delas em ícones.

Obrigado !




Até a próxima...

Apresentador

Thales Levi Azevedo Valente

E-mail:
thales.l.a.valente@gmail.com

Referências

- Links referenciados nos respectivos slides.
- T.B. Borchartt . *Introdução à Inteligência Artificial*. 2024. 37 slides. Universidade Federal do Maranhão.
- A.O. B. Filho. *Inteligência Artificial - Introdução*. 2024. 31 slides. Universidade Federal do Maranhão.