

# Aula 02 - Métodos de Resolução de Problemas e Busca em Espaço de Estados.

Profa. Gabrielly Queiroz

# Contextualização

- Computadores não são inteligentes, pois apenas executam as instruções que os programadores escrevem.
- Se um computador pudesse, por conta própria, **criar suas próprias regras e decidir o que fazer sem precisar de instruções pré-definidas**, ele seria considerado mais inteligente.
- Esse é um dos grandes desafios da Inteligência Artificial: dar às máquinas a capacidade de **auto programação**, ou seja, de formular suas próprias soluções para problemas.
- Uma das primeiras técnicas desenvolvidas para isso foi a **busca em espaço de estados**. Apesar de ser uma abordagem básica, ela é **fundamental para diversas outras áreas da IA**.
- O espaço de estados ensina a máquina a **explorar diferentes possibilidades e encontrar soluções**, imitando um processo de pensamento lógico.

# Contextualização

Problema → Criamos um programa (Python, Java, etc.) → O computador executa → Problema resolvido.

**A inteligência está no programador**, não no computador.

O computador apenas segue as instruções, sem tomar decisões por conta própria.

Em vez de programar a solução diretamente, criamos um **agente** que busca a solução.

**O agente** interage com o ambiente e toma decisões.

Ele explora diferentes possibilidades até alcançar um **objetivo**.

A inteligência agora está no processo de busca, e não apenas no código fixo

Um problema pode ser representado por um **espaço de estados**.

O agente navega nesse espaço até encontrar a solução.

## Elementos principais:

- **Estados**: Diferentes configurações do problema.
- **Ações**: Como o agente pode mudar de um estado para outro.
- **Objetivo**: O estado final desejado.

# Busca em Espaço de Estados

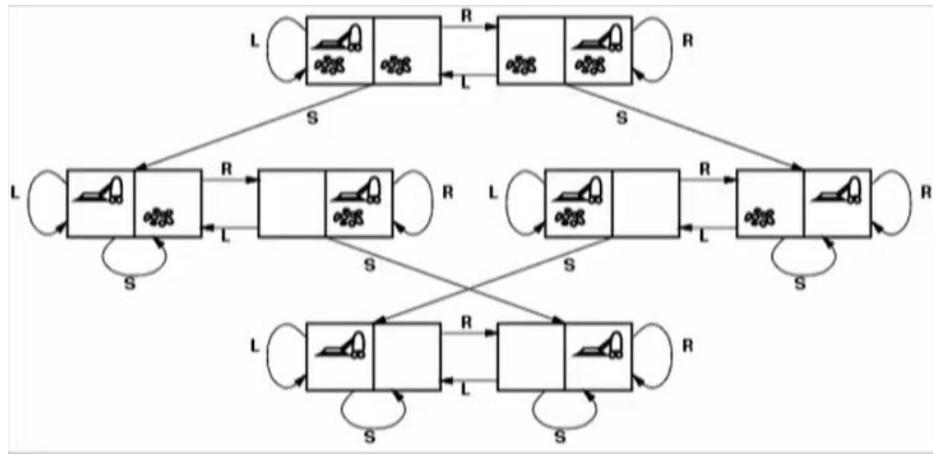
- Abordagem de Busca: Agente orientada a meta/objetivo.
- O agente está dentro do problema, tem o objetivo que quer atingir e tem o que ele percebe.
- O processo produz um programa de computador que vai resolver o problema.
- Executa o programa e resolve.

# Exemplo do aspirador de pó

- Um robô aspirador de pó deve limpar uma casa com duas posições.  
Operações que ele sabe executar:
  - Sugar.
  - Ir para a posição da esquerda.
  - Ir para a posição da direita.
- Como o aspirador pode montar um plano para limpar a casa se inicialmente ele está na posição direita e as duas posições tem sujeira?
- Quais os estados possíveis do mundo do aspirador e as transições?

# Exemplo do aspirador de pó

- Busca: Ações e Estados.
- Ações: Sugar. Esquerda. Direita
- Estados: 2 posições. Limpo ou sujo.
- Cenários: casa suja e posição esquerda. Casa suja e na direita. Casa limpa e na direita. Casa limpa e na esquerda. Sujeira apenas na direita e robo na esquerda. Sujeira apenas na direita e robo na direita. Sujeira apenas na esquerda e robo na esquerda. Sujeira apenas na direita e robo na esquerda.
- Transformações de estados com base nas ações. Grafos.
- Estado Atual e Objetivo.
- Programação: Achar um caminho pelo grafo.



# O que é busca?

- O mundo do agente é modelado por conjunto de estados possíveis (muitas vezes este conjunto é infinito)
- Existem transições entre os estados do mundo, formando um grafo
- São utilizados algoritmos para encontrar um caminho neste grafo
  - Partindo do estado inicial (atual)
  - até o estado objetivo

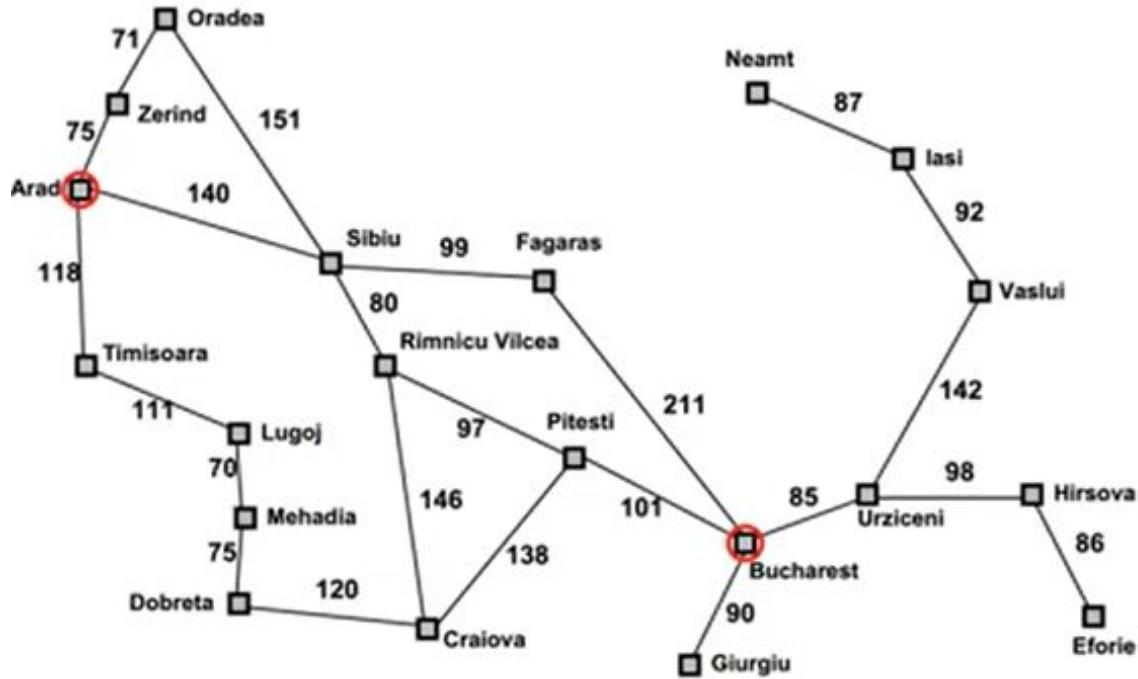
# Problema de Busca

- Busca:
  - Estados - Atual e Objetivo
  - Ações
- Problema de busca: achar uma sequência de ações que leva o estado atual ao estado objetivo.
- Um agente com várias ações imediatas pode decidir o que fazer comparando diferentes sequências de ações possíveis.
- Esse processo de procurar pela melhor sequência é chamado de busca
- Formular objetivo - Buscar - Executar

# Exemplo: férias na Romênia

De férias na Romênia;  
atualmente em Arad.

Voo sai amanhã de Bucareste



## Exemplo: férias na 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.

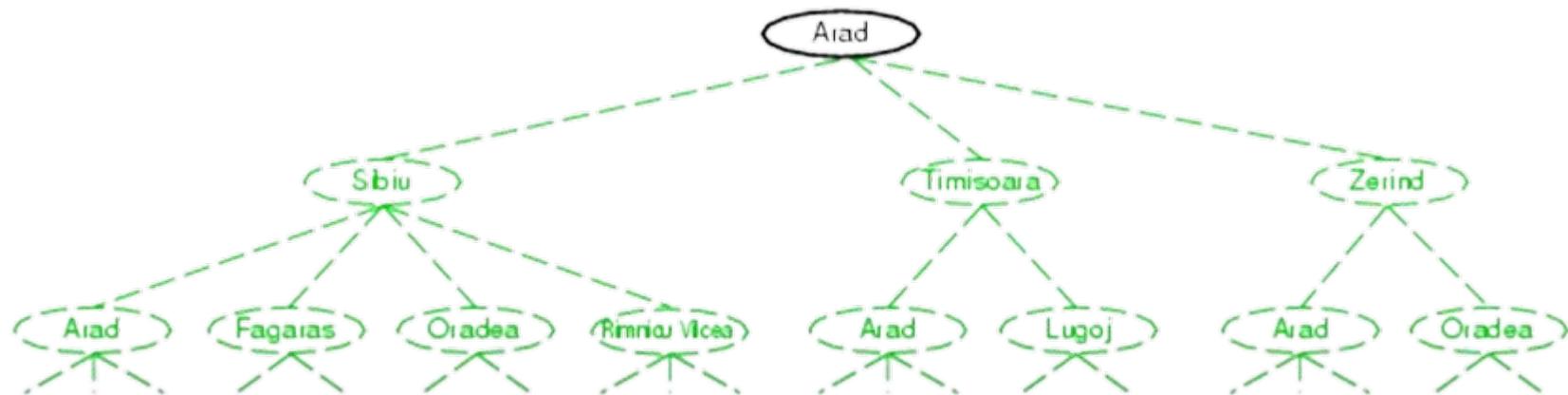
# Formulação de Problemas

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

# Busca de Soluções

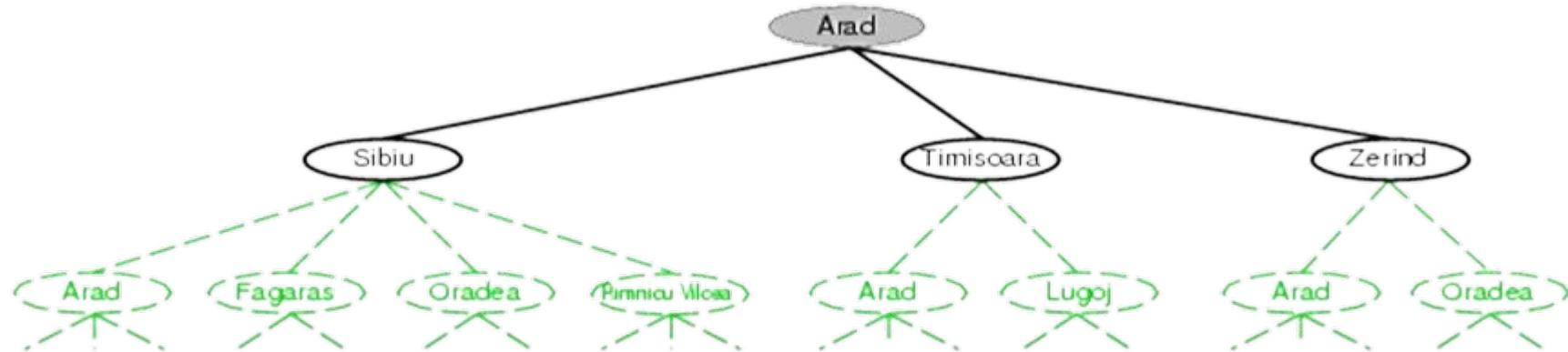
- Ideia: percorrer o espaço de estados a partir de uma árvore de busca.
- Expandir o estado atual aplicando a função sucessor, gerando novos estados.
- Busca: seguir um caminho, deixando os outros para depois.
- A estratégia de busca determina qual caminho seguir.

# Exemplo de árvore de busca



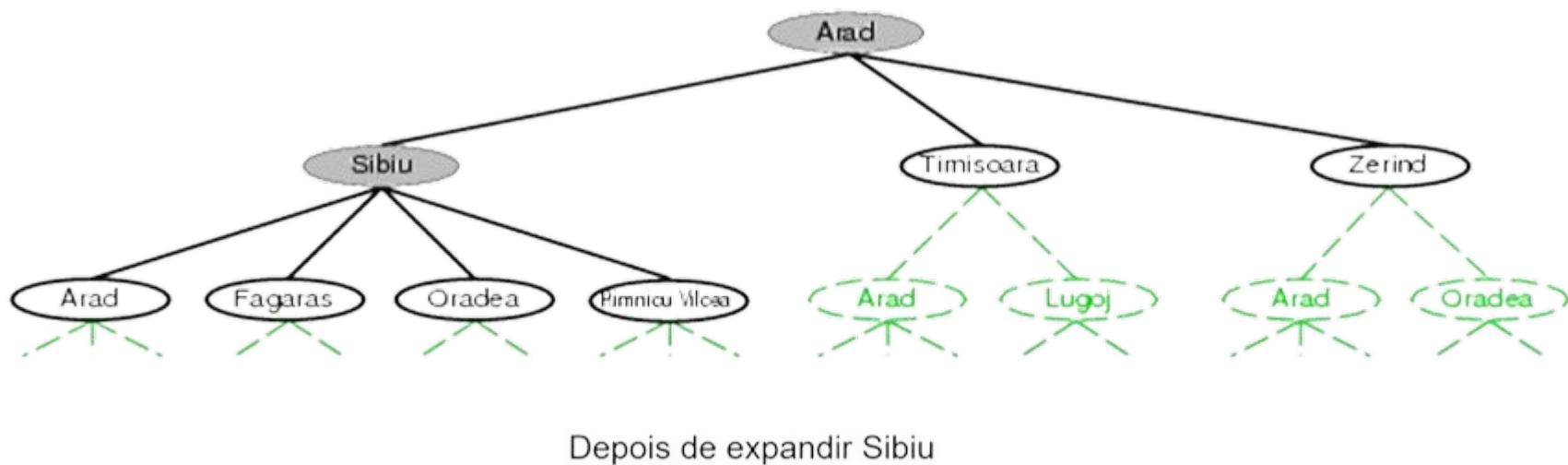
Estado inicial

# Exemplo de árvore de busca



Depois de expandir Arad

# Exemplo de árvore de busca



# Termos importantes

**Agente** → Quem toma decisões e executa ações no ambiente.

**Estado** → Uma configuração específica do problema em um momento.

**Espaço de estados** → Todos os estados possíveis que o agente pode explorar.

**Árvore de busca** → Representação dos caminhos que o agente pode seguir para resolver o problema.

**Nó de busca** → Cada ponto na árvore de busca que representa um estado possível.

**Objetivo/Meta** → O estado final desejado que o agente precisa alcançar.

**Ação** → O movimento que o agente pode fazer para mudar de estado.

**Função sucessor** → Define quais estados novos podem ser alcançados a partir de um estado atual.

# Por que espaço de estados é importante para IA?

**Modela problemas** → Permite representar qualquer problema como um conjunto de estados e transições.

**Ajuda na tomada de decisão** → O agente pode explorar diferentes possibilidades antes de agir.

**Permite busca de soluções** → A IA pode encontrar um caminho ótimo sem precisar de um código fixo para cada problema.

**Base para algoritmos de busca** → Métodos como BFS e DFS usam o espaço de estados para encontrar soluções.

**Aplicável em diversas áreas** → Usado em jogos, robótica, planejamento de rotas e sistemas de diagnóstico.

**Facilita a automação** → A IA pode resolver problemas sem intervenção humana direta.

# Algoritmos de Busca

- **Busca Cega (Sem Inteligência)**
  - **Não usa heurísticas** → Explora todas as possibilidades sem priorizar caminhos melhores.
  - **Segue regras fixas** → Pode ser lento e ineficiente para problemas grandes.
- 
- **Busca Informada (Com Inteligência)**
  - **Usa heurísticas** → Estima quais caminhos têm mais chance de levar à solução.
  - **Torna a busca mais eficiente** → Evita explorar estados desnecessários.