

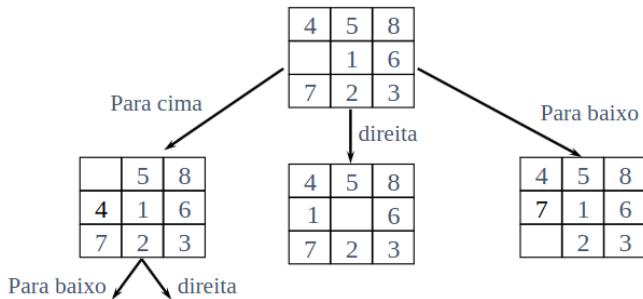
# Inteligência Artificial

Guilherme Henrique de Souza Nakahata

Universidade Estadual do Paraná - Unespar

18 de Abril de 2024

# Busca cega



# Busca com aprofundamento iterativo e limitada

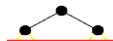
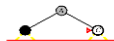
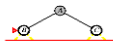
- Busca em profundidade limitada:
  - Resolve o problema de busca em profundidade em árvores infinitas;
  - Evita o problema de caminhos muito longos ou infinitos;
  - Impõe um limite máximo de profundidade para os caminhos gerados;
- Busca em profundidade iterativo:
  - Combina busca em profundidade com busca em largura;
  - Faz a busca em profundidade aumentando gradualmente o limite de profundidade;
  - Ideal quando se tem espaço de busca grande e profundidade não conhecida;
  - Limites com valores crescentes;
  - Partindo do zero;
  - Encontrar a primeira solução.

# Busca com aprofundamento iterativo

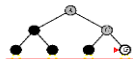
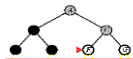
Limit = 0



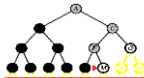
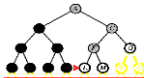
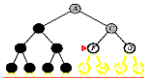
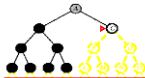
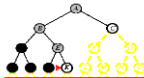
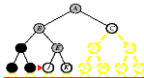
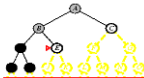
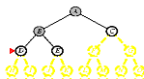
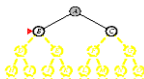
Limit = 1



Limit = 2



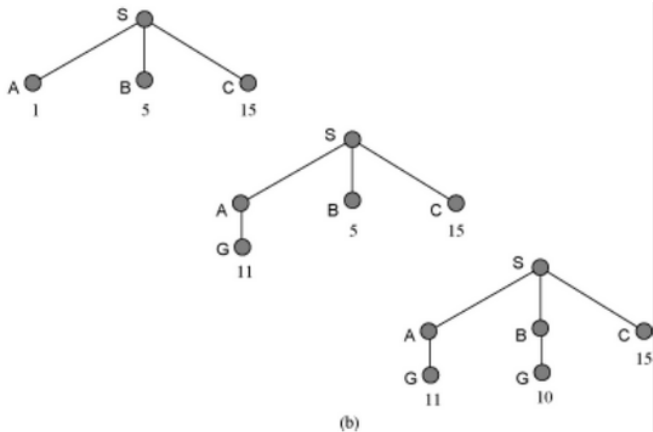
Limit = 3



# Busca de custo uniforme

- Modifica a busca em largura:
  - Expande o nó da fronteira com menor custo de caminho;
  - Na fronteira do espaço de estados;
  - Cada operador pode ter um custo associado diferente;
  - Medido pela função  $g(n)$ , para o nó  $n$ ;
  - $g(n)$  é o custo do caminho da origem ao nó  $n$ .
- Na busca em largura:  $g(n) = \text{profundidade}(n)$ .

# Busca de custo uniforme



# Busca de custo uniforme

- $F = S$ 
  - testa se  $S$  é o estado objetivo, expande-o e guarda seus filhos  $A$ ,  $B$  e  $C$  ordenadamente na fronteira
- $F = \{A, B, C\}$ 
  - testa  $A$ , expande-o e guarda seu filho  $G_a$  ordenadamente
  - obs: o algoritmo de geração e teste guarda na fronteira todos os nós gerados, testando se um nó é o objetivo apenas quando ele é retirado da lista!
- $F = \{B, G_a, C\}$ 
  - testa  $B$ , expande-o e guarda seu filho  $G_B$  ordenadamente
- $F = \{G_b, G_a, C\}$ 
  - testa  $G_b$  e para!

- Projeto Shakey;
- Robô móvel;
- Planejar suas próprias ações;
- Graph traverser;
- Função heurística  $h(n)$ ;
- Estimativa do custo do caminho restante até o objetivo;
- Orientar a busca em direção e regiões mais promissoras no espaço de busca.



- Admissibilidade;
  - Nunca superestima o custo para alcançar o objetivo;
  - Estimativa de custo nunca for maior do que o custo real;
- Consistência;
  - A estimativa de custo de um estado nunca é aumentada ao longo do caminho até o objetivo;
- A\*;
- Caminhos de menor custo;
- Caminhos ótimos (Álgebra de custos)

- Aplicações no mundo real;
- Jogos;
- Navegação de GPS;
- Movimento de robôs;
- Planejamento de rotas em logística;
- Design de circuitos integrados.

- Menor custo de deslocamento de um ponto de origem A até o destino B;
- Fórmula de custo tem uma combinação total dada por:
  - $f(x) = g(x) + h(x)$ ;
  - $g(x)$  = função de custo sobre uma posição de origem até a posição;
  - $h(x)$  = Função heurística.
- 8-puzzle  $h(x)$  é a distância de Manhattan entre a posição atual de x e sua posição final;
- Lista aberta;
- Lista fechada;

# Algoritmo A\*

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>
<b>K</b>	<b>L</b>	<b>M</b>	<b>N</b>	<b>O</b>
<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>
<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>

# Algoritmo A\*

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

# Algoritmo A\*

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

# Obrigado! Dúvidas?

Guilherme Henrique de Souza Nakahata

[guilhermenakahata@gmail.com](mailto:guilhermenakahata@gmail.com)

<https://github.com/GuilhermeNakahata/UNESPAR-2024>