

# Inteligência Artificial

Busca sem informação



Prof. Chauã Queirolo



<https://github.com/chaua/inteligencia-artificial>

# Sumário

- Busca sem informação

# Introdução

# Introdução

- Um problema pode ser definido por 5 componentes
  - Estado inicial
  - Ações
  - Modelo de transição
  - Teste de objetivo
  - Custo do caminho

# Introdução

## Solução

- Sequência de ações que levam do **estado inicial** para o **estado objetivo**

## Solução ótima

- Solução com o menor **custo de caminho**

# Algoritmo geral de busca em árvore

```
function TREE-SEARCH(problem, fringe) returns a solution, or failure  
  fringe  $\leftarrow$  INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)  
  loop do  
    if fringe is empty then return failure  
    node  $\leftarrow$  REMOVE-FRONT(fringe)  
    if GOAL-TEST[problem](STATE[node]) then return SOLUTION(node)  
    fringe  $\leftarrow$  INSERTALL(EXPAND(node, problem), fringe)
```

---

```
function EXPAND(node, problem) returns a set of nodes  
  successors  $\leftarrow$  the empty set  
  for each action, result in SUCCESSOR-FN[problem](STATE[node]) do  
    s  $\leftarrow$  a new NODE  
    PARENT-NODE[s]  $\leftarrow$  node; ACTION[s]  $\leftarrow$  action; STATE[s]  $\leftarrow$  result  
    PATH-COST[s]  $\leftarrow$  PATH-COST[node] + STEP-COST(node, action, s)  
    DEPTH[s]  $\leftarrow$  DEPTH[node] + 1  
    add s to successors  
  return successors
```

**Busca** sem informação

# Busca sem informação

- Estratégias de **busca sem informação** usam apenas a **informação** disponível na definição do **problema**
- Apenas geram **sucedores** e verificam se o **estado objetivo** foi atingido

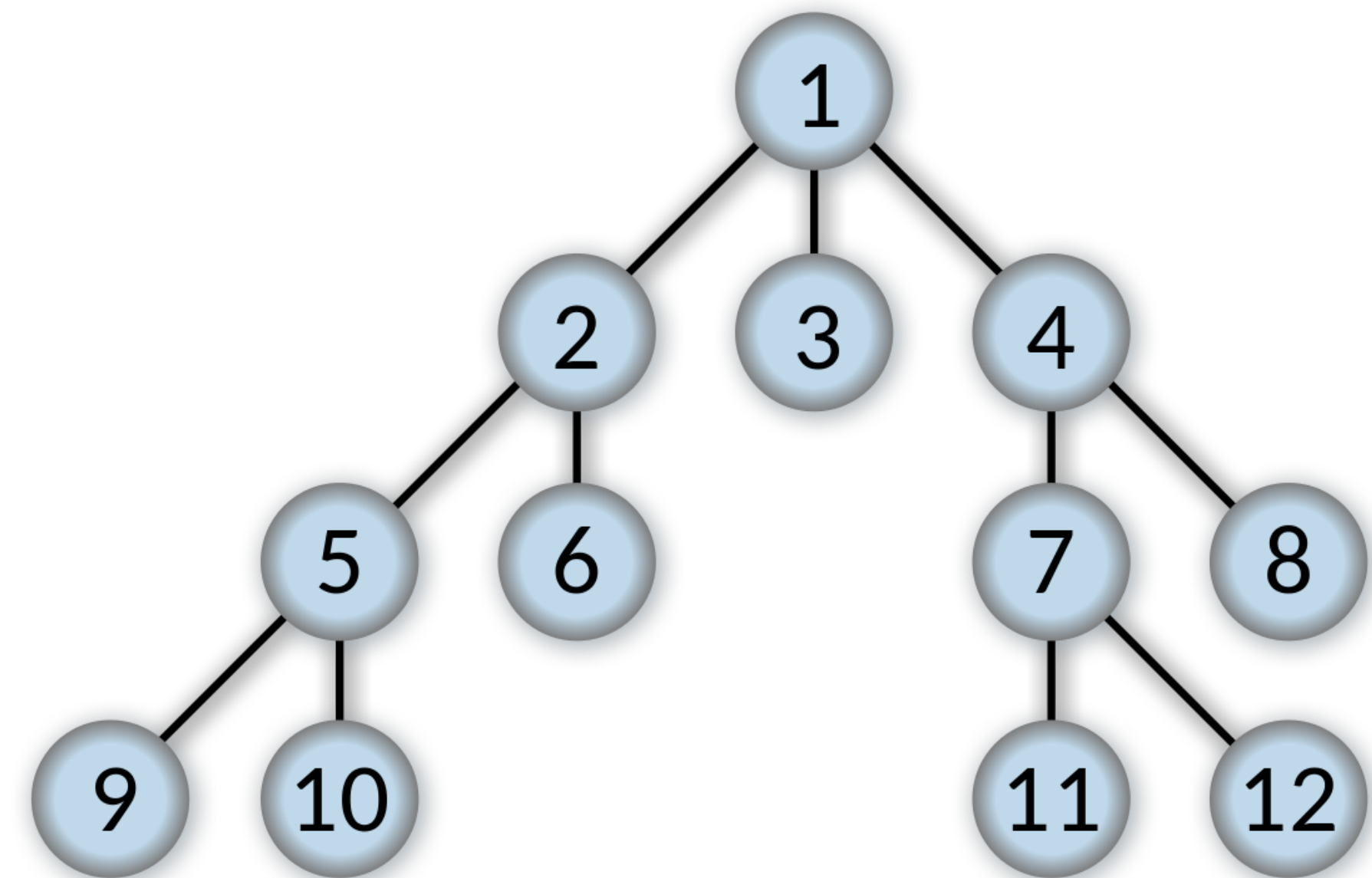


# Busca sem informação

- As estratégias de busca sem informação se distinguem pela **ordem** em que os **nós** são **expandidos**
  - Busca em largura (Breadth-first)
  - Busca de custo uniforme
  - Busca em profundidade (Depth-first)
  - Busca em profundidade limitada
  - Busca de aprofundamento iterativo

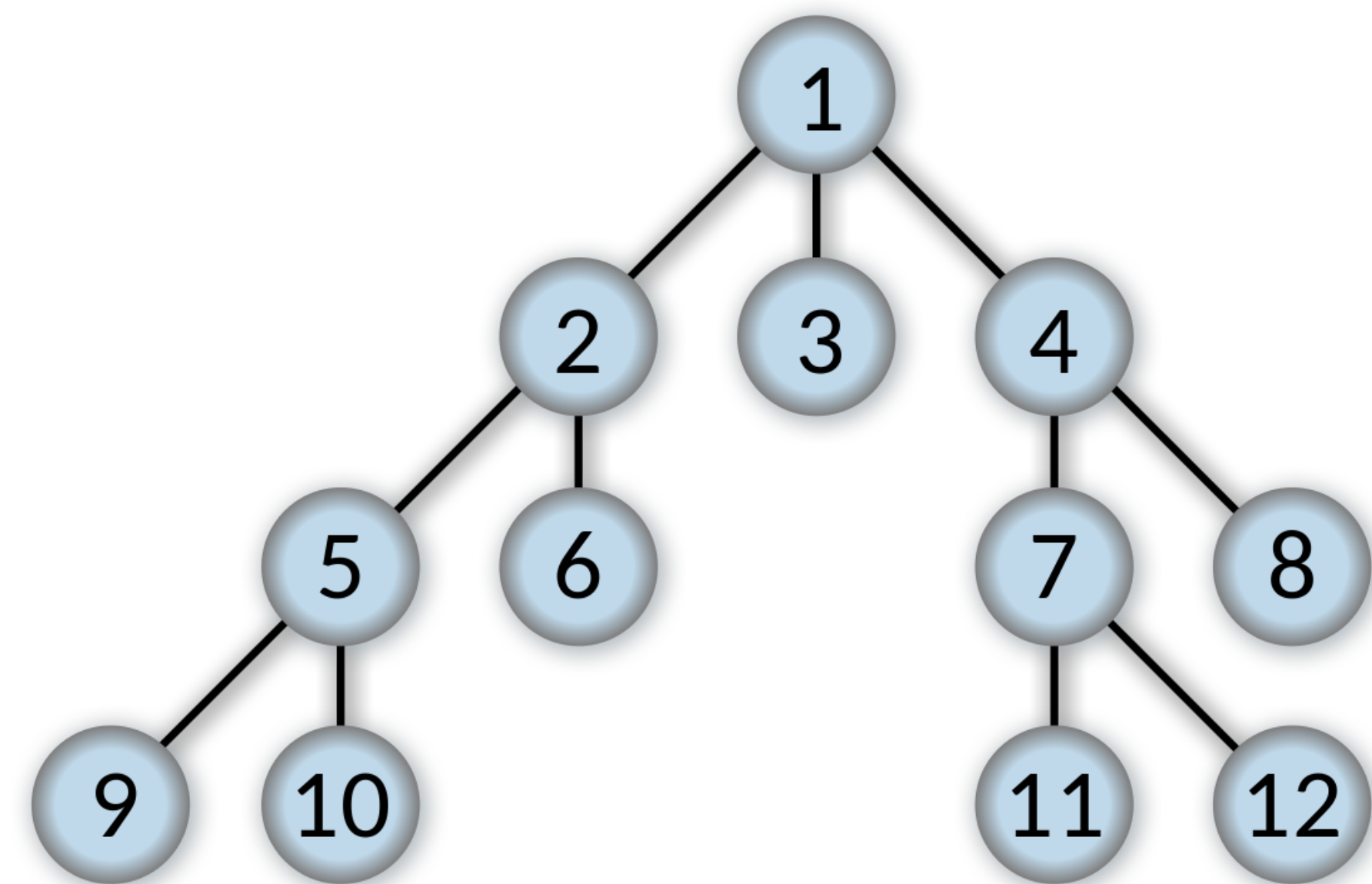
# Busca em largura

- Expandir o nó não-expandido mais perto da raiz
- **Implementação**
  - A borda é uma fila FIFO
  - Novos itens entram no final.



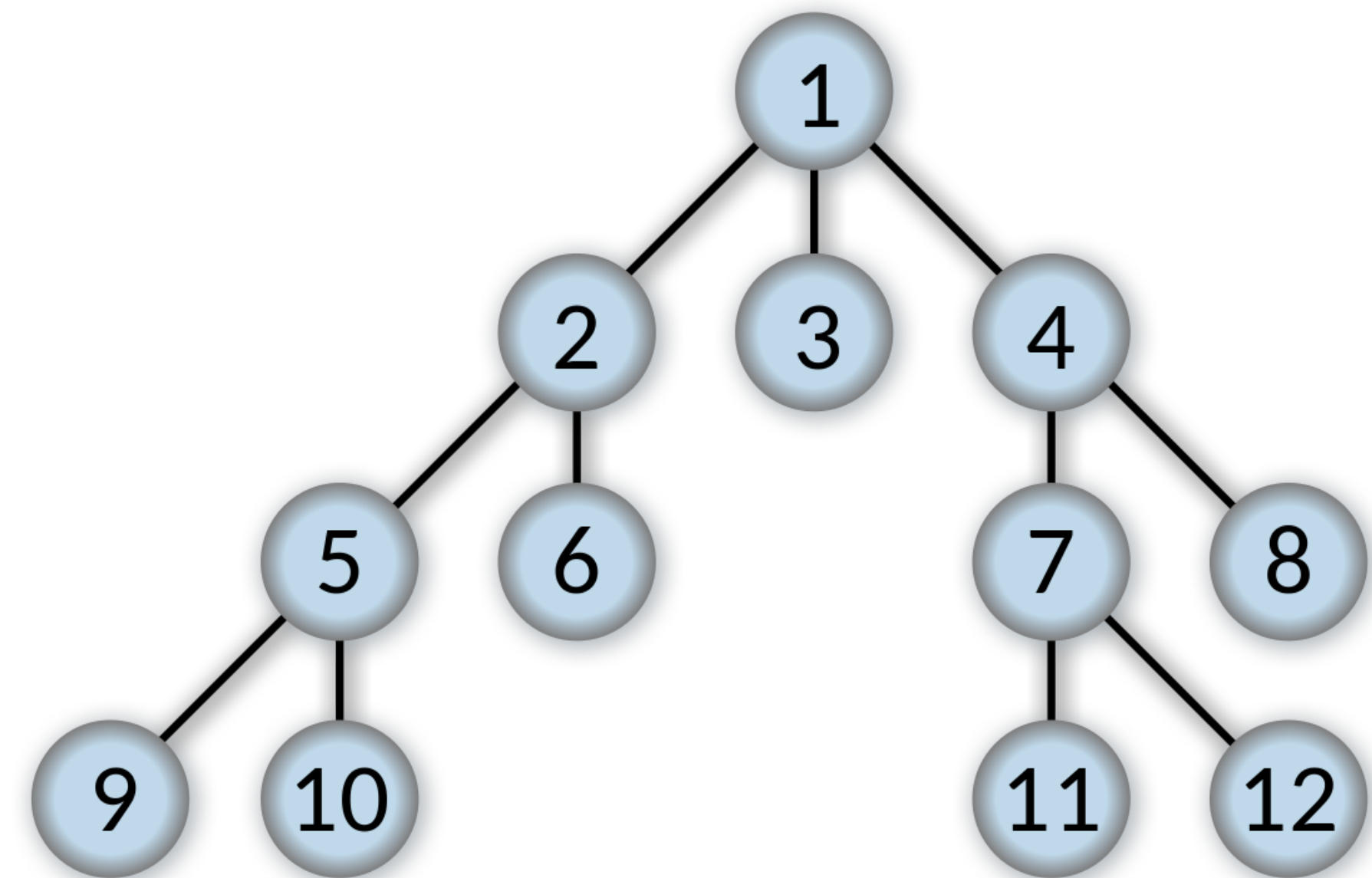
# Busca em profundidade

- Expandir o nó não-expandido mais profundo
- **Implementação**
  - A borda é uma fila LIFO
  - Uma pilha



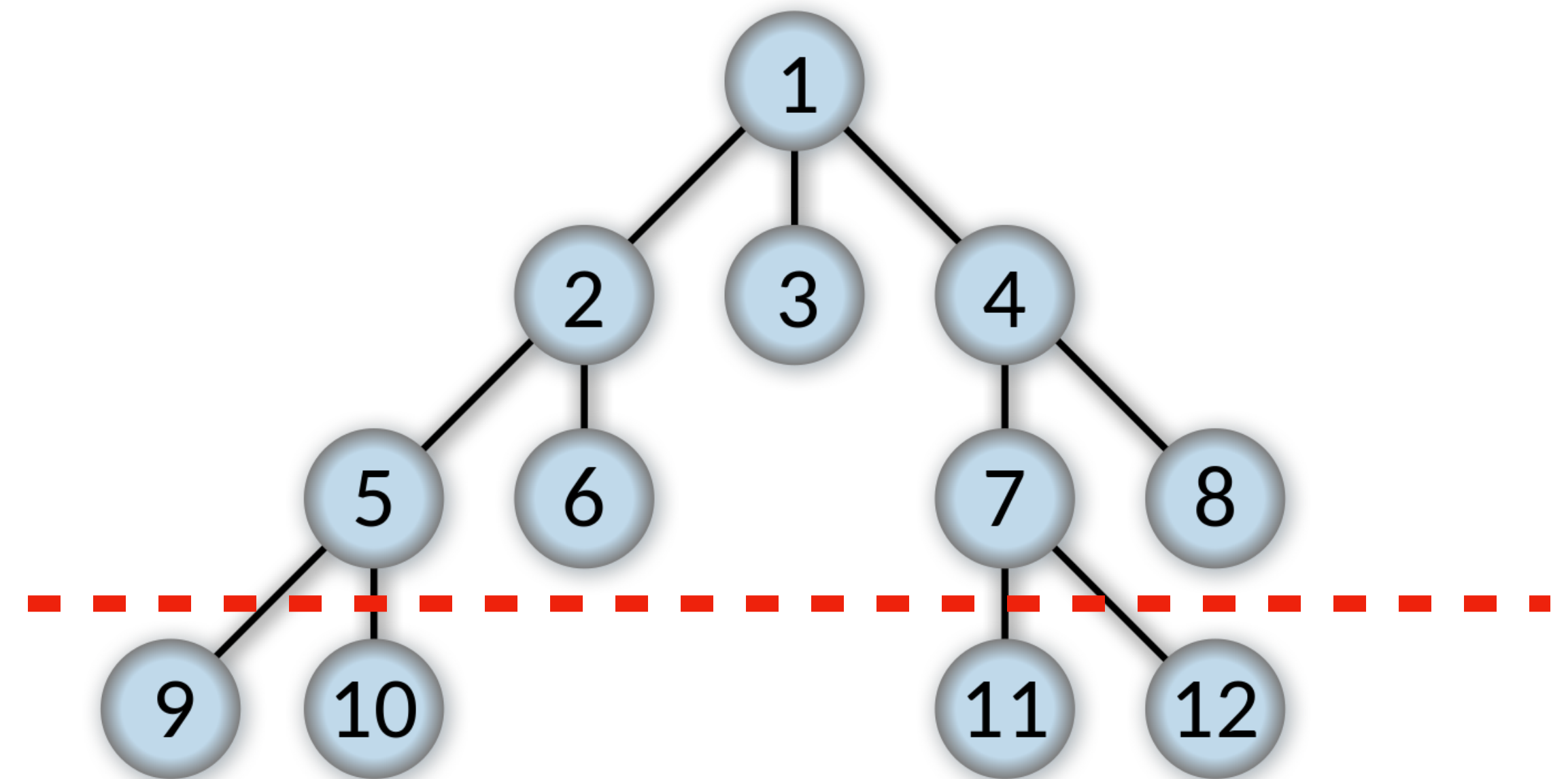
# Busca de custo uniforme

- Expandir o nó não-expandido que tenha o custo de caminho mais baixo
- **Implementação**
  - A borda é uma heap



# Busca em profundidade limitada

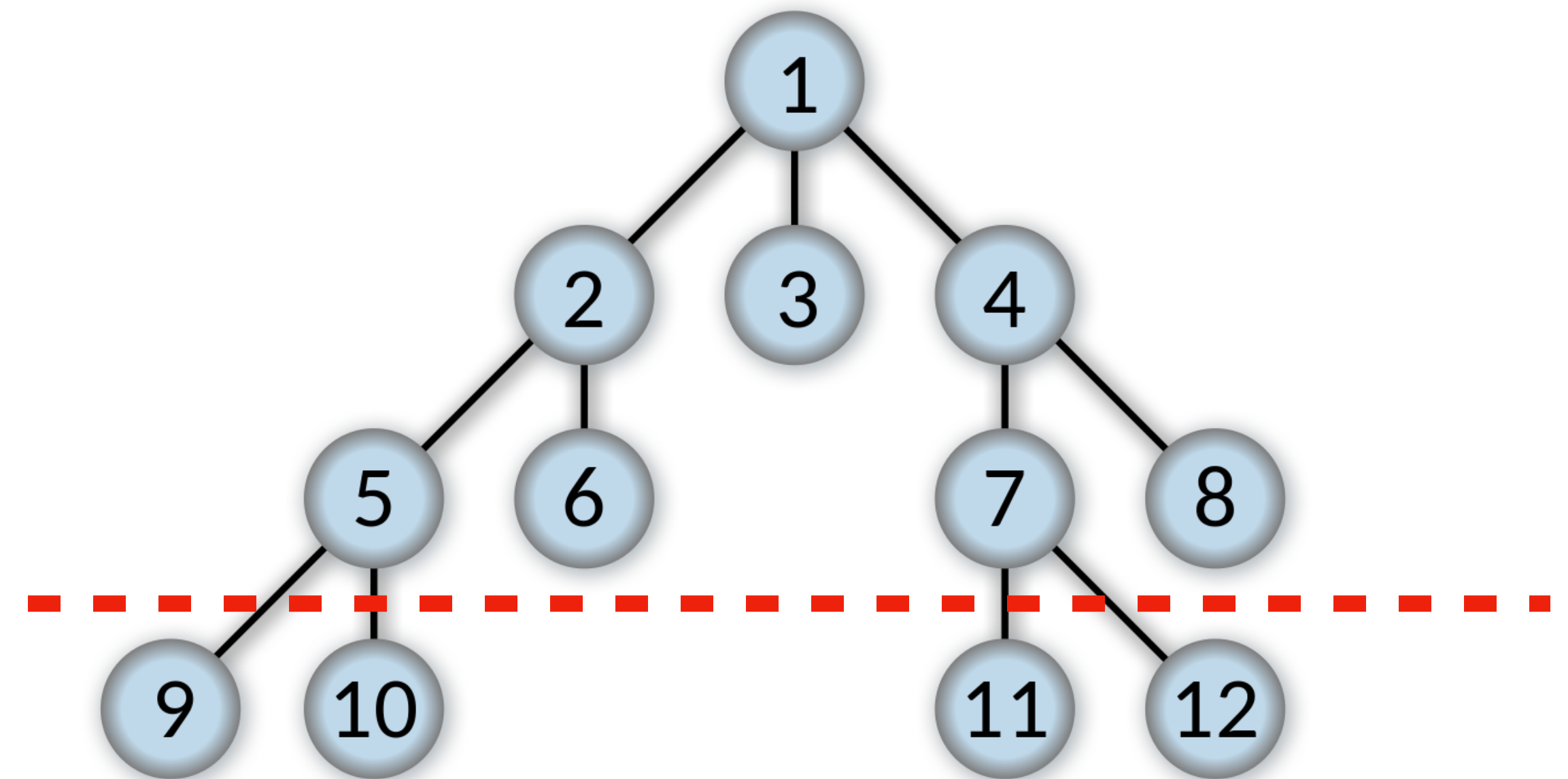
- Realiza busca em profundidade até uma altura definida





# Busca de aprofundamento iterativo

- Realiza busca em profundidade em níveis de altura pré-definidos



# Estados repetidos

- O **processo de busca** pode perder tempo expandindo nós já **explorados**
  - Estados repetidos podem levar a **loops infinitos**
  - Estados repetidos podem transformar um **problema linear** em um **problema exponencial**

# Estados repetidos

- **Comparar** os nós prestes a serem **expandidos** com nós já **visitados**.
  - Se o nó já tiver sido visitado, será descartado
  - Lista armazena nós já visitados
- Busca em profundidade e busca de aprofundamento iterativo não tem mais espaço linear
  - A busca percorre um grafo e não uma árvore



# Referências

## Bibliográficas



# Referências Bibliográficas

- S. J. Russell & P. Norvig. **Artificial Intelligence: A Modern Approach.** Prentice Hall, 3rd edition, 2010.