



Abordagens para Resolução de Problemas

UDESC - Universidade do Estado de Santa Catarina
Departamento de Ciência da Computação

CAL - Complexidade de Algoritmos

Teodoro Alberto Borges Junior

Abordagens para Resolução de Problemas

Muitos problemas podem ser resolvidos de diversas maneiras diferentes.

Contudo, dependendo da abordagem escolhida, uma solução pode ser melhor ou pior do que outra em questões de tempo ou uso de memória.

Abordagens para Resolução de Problemas

Podemos classificar os principais métodos de resolução de problemas em relação a abordagem utilizada:

- Indução matemática - fórmula
- Divisão e Conquista (*divide and conquer*)
- Algoritmos Gulosos (*greedy*)
- Algoritmo de Tentativa e Erro (*backtracking*)
- Programação Dinâmica (*dynamic programming*)
- Algoritmos de Aproximação (*approximation*)

Tentativa e Erro (*backtracking*)

Testa todas as possíveis soluções até encontrar a(s) melhor(es) solução(ões) para um problema, utilizando as ideias de busca em profundidade ou busca em largura.

Tentativa e Erro (*backtracking*)

Testa todas as possíveis soluções até encontrar a(s) melhor(es) solução(ões) para um problema, utilizando as ideias de busca em profundidade ou busca em largura.

Pode utilizar validações para diminuir o escopo de busca.

Geralmente não é uma abordagem eficaz para instâncias muito grandes.

Tentativa e Erro (*backtracking*) – Força Bruta

Exemplo: mostrar todas as combinações binárias possíveis com n dígitos.

$n = 2$

00

01

10

11

$n=3$

000

100

001

101

010

110

011

111

Tentativa e Erro (*backtracking*) – Força Bruta

Exemplo: mostrar todas as combinações binárias possíveis com n dígitos.

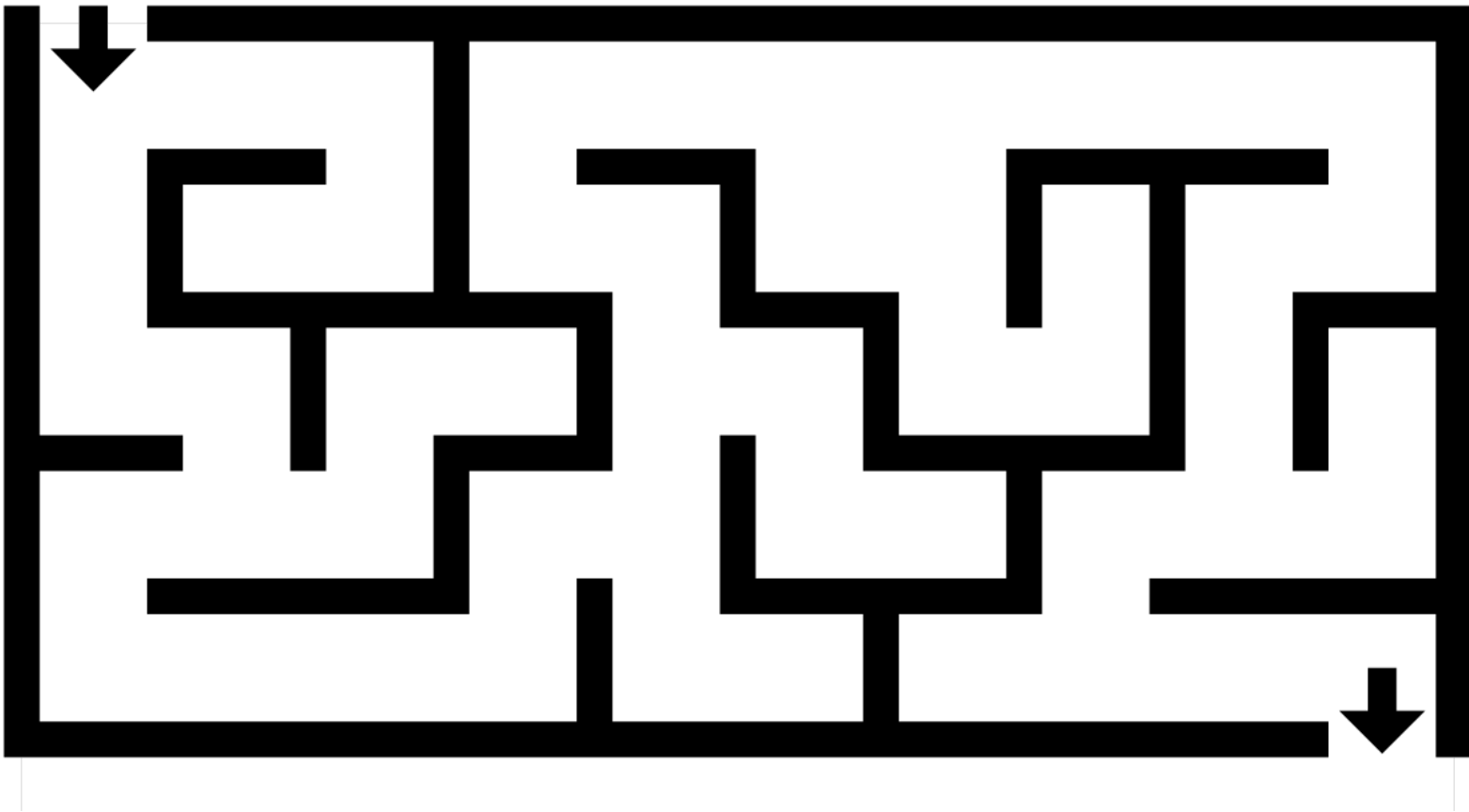
```
prefixo = ""  
mostraDigitos(n, prefixo)  
    se (n == 0)  
        print(prefixo)  
    senão  
        mostraDigitos(n-1, prefixo + "0")  
        mostraDigitos(n-1, prefixo + "1")
```

Tentativa e Erro (*backtracking*)

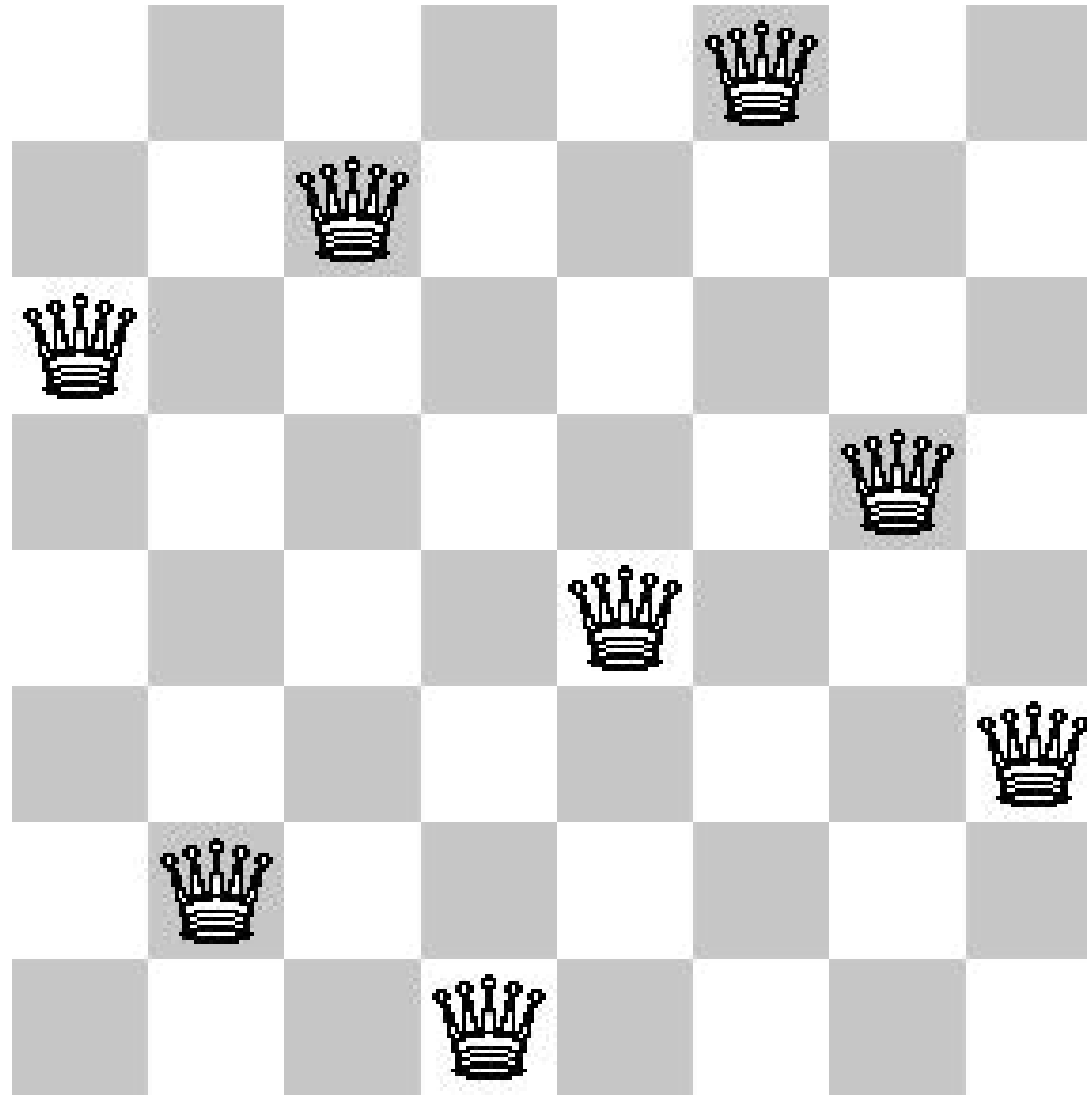
Alguns problemas clássicos que podem ser resolvidos com *backtracking*:

- Labirinto (*maze*)
- *N Queens*
- Jogo-da-velha (*tic-tac-toe*)
- Puzzle 3x3
- Sokoban
- Caixeiro Viajante

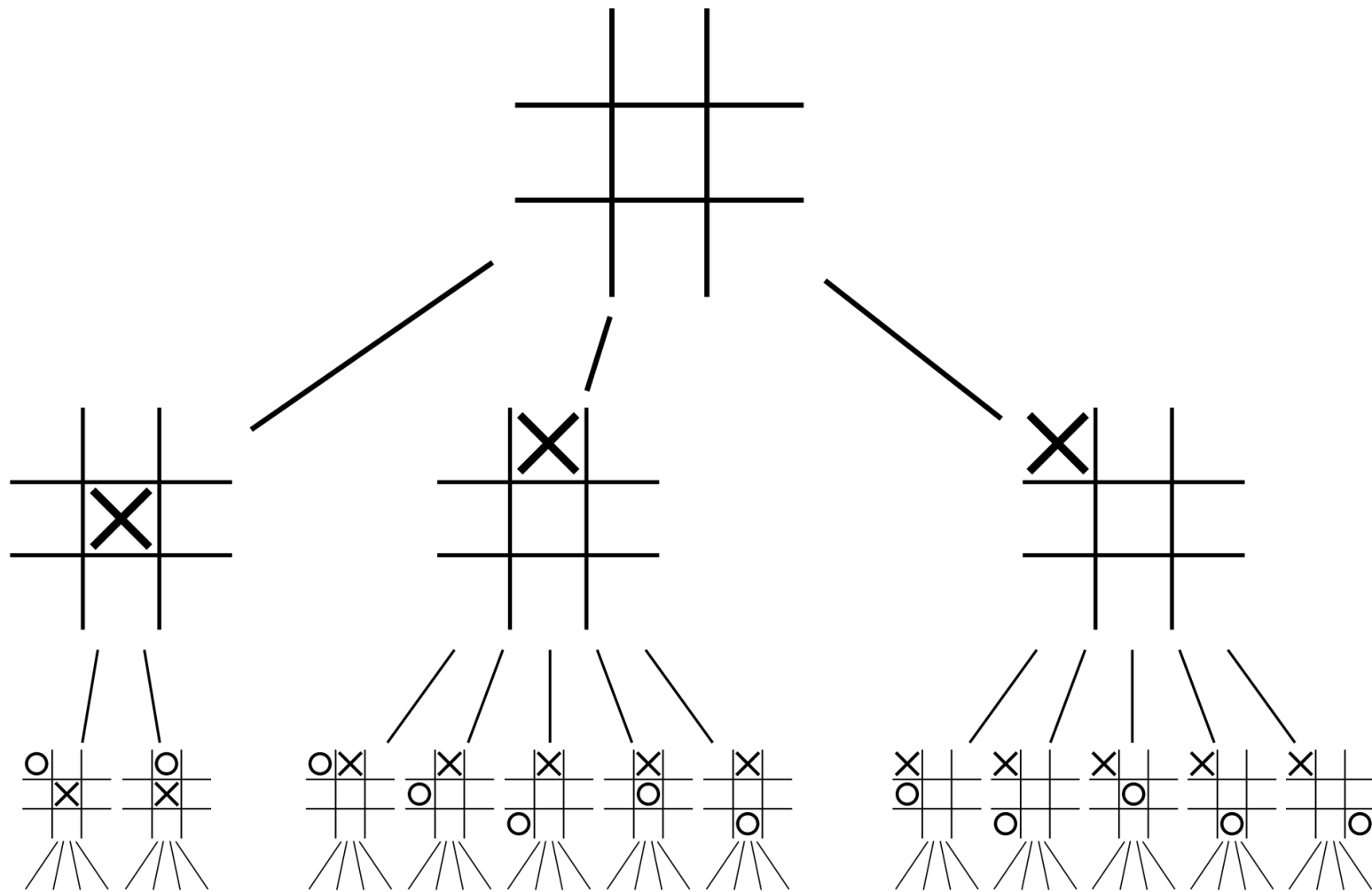
Tentativa e Erro (*backtracking*) - Labirinto



Tentativa e Erro (*backtracking*) – *N* Queens



Tentativa e Erro (*backtracking*) – *Tic-Tac-Toe*



Tentativa e Erro (*backtracking*) – Puzzle 3x3

EXERCÍCIO:

Escreva um programa que informa qual o número mínimo de jogadas para resolver o seguinte Puzzle 3x3

1	0	3
4	2	5
7	8	6

Linguagens: C / C++ / C# / Java / Python / Haskell