

# MAC121 - Algoritmos e Estruturas de Dados I

Universidade de São Paulo

Segundo Semestre de 2020

## Aplicação de pilhas - backtrack

## Backtrack - problema das 8 rainhas

**Problema:** É possível colocar, em um tabuleiro de xadrez, 8 rainhas sem que elas se ataquem?



O problema foi proposto em 1848 por Max Bezzel, e a primeira solução apareceu em 1850, devida a Frank Nauck, que generalizou o problema para um tamanho qualquer de tabuleiro.

## O problema das $n$ rainhas

Veja a página na Wikipedia para mais informações:

- ▶ Para  $n = 1$ , temos uma solução.
- ▶ Para  $n = 2$  e  $n = 3$ , não é possível.
- ▶ Para  $n = 4$ :

|   |   |   |   |
|---|---|---|---|
|   | R |   |   |
|   |   |   | R |
| R |   |   |   |
|   |   | R |   |

São duas soluções simétricas.

- ▶ Para  $n = 8$  temos 92 soluções, das quais, 12 são “diferentes”.

## Problema das $n$ rainhas

Para  $n = 8$  teríamos que testar  $8! = 40320$  possíveis jeitos de colocar as 8 rainhas, uma por linha e coluna. Mas, como fazer isso?

A ideia é, para cada rainha, tentar colocá-la na primeira coluna que é possível. Se conseguimos colocar todas as rainhas, encontramos uma solução. Caso contrário, **retornamos à última rainha colocada** e tentamos mudar esta decisão, e seguimos a partir daí.

Esta estratégia é chamada **backtrack**, e armazenamos as decisões tomadas em uma pilha!

## Backtrack - esquema geral

```
enquanto não colocamos as  $n$  rainhas
  procura uma coluna livre na linha atual
  se achou
    empilha esta decisão
    coloca a rainha nesta posição
    segue para a próxima linha
  senão /* backtrack */
    se a pilha está vazia
      Não tem solução!
    senão
      Desempilha a última decisão
      Volta à rainha anterior
      Tira da coluna que estava
      Volta a tentar a partir da próxima coluna
```

Como gerar todas as soluções?