

# **Algoritmo Genético e Busca tabu no problema n-rainhas**

## **Trabalho 1**

**Sergio A. – RA: 115735  
Gabriel T. – RA: 107774**

**Prof. wagner Igarashi**

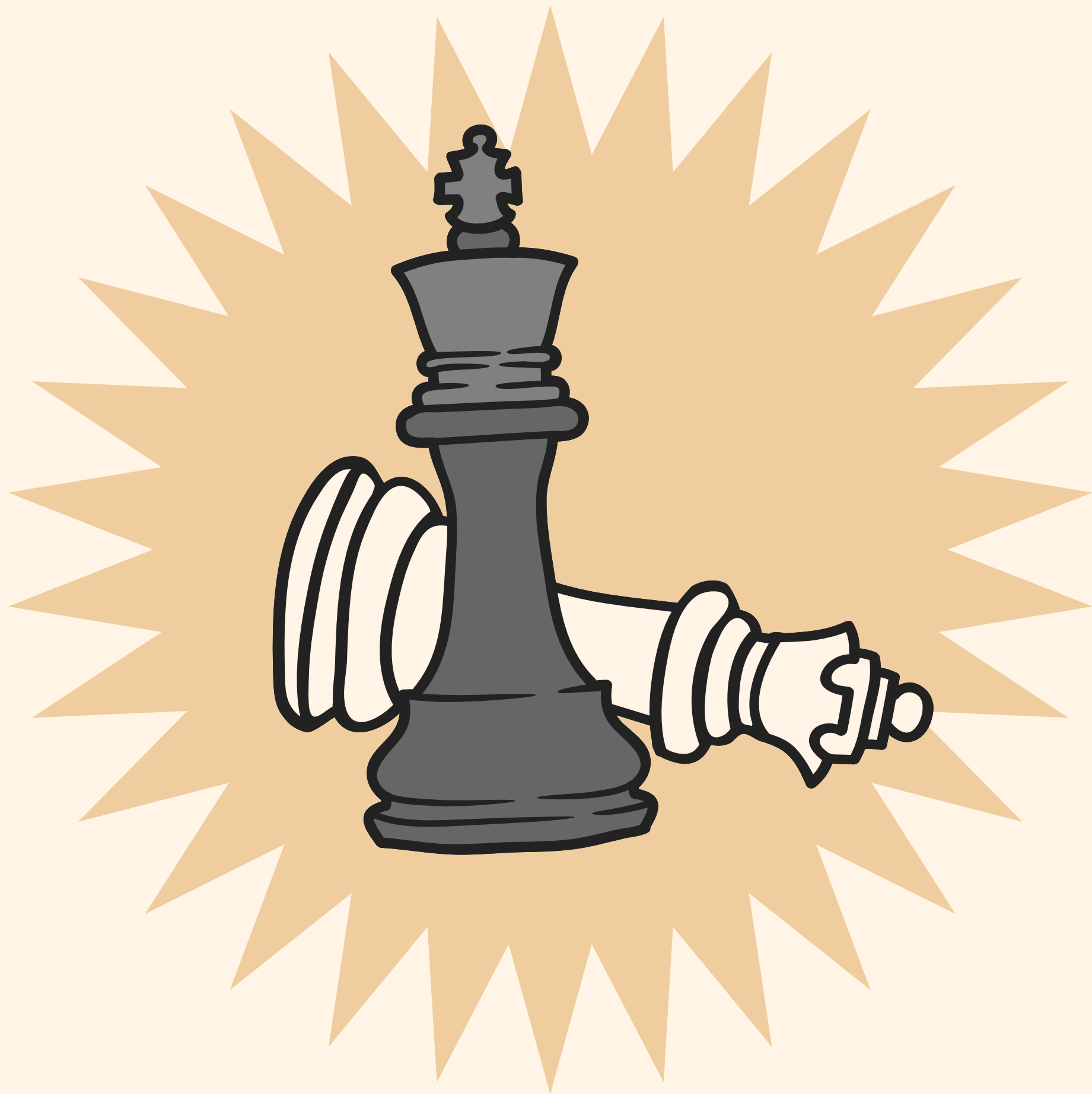
# SUMÁRIO

**01 Problema**  
Explicação do problema  
escolhido

**02 Técnicas**  
SObre as 2 técnica de  
resolução

**03 Explicações**  
Detalhes sobre os  
códigos das técnicas

**04 Resultados**  
Juntamente com os  
gráficos resultantes



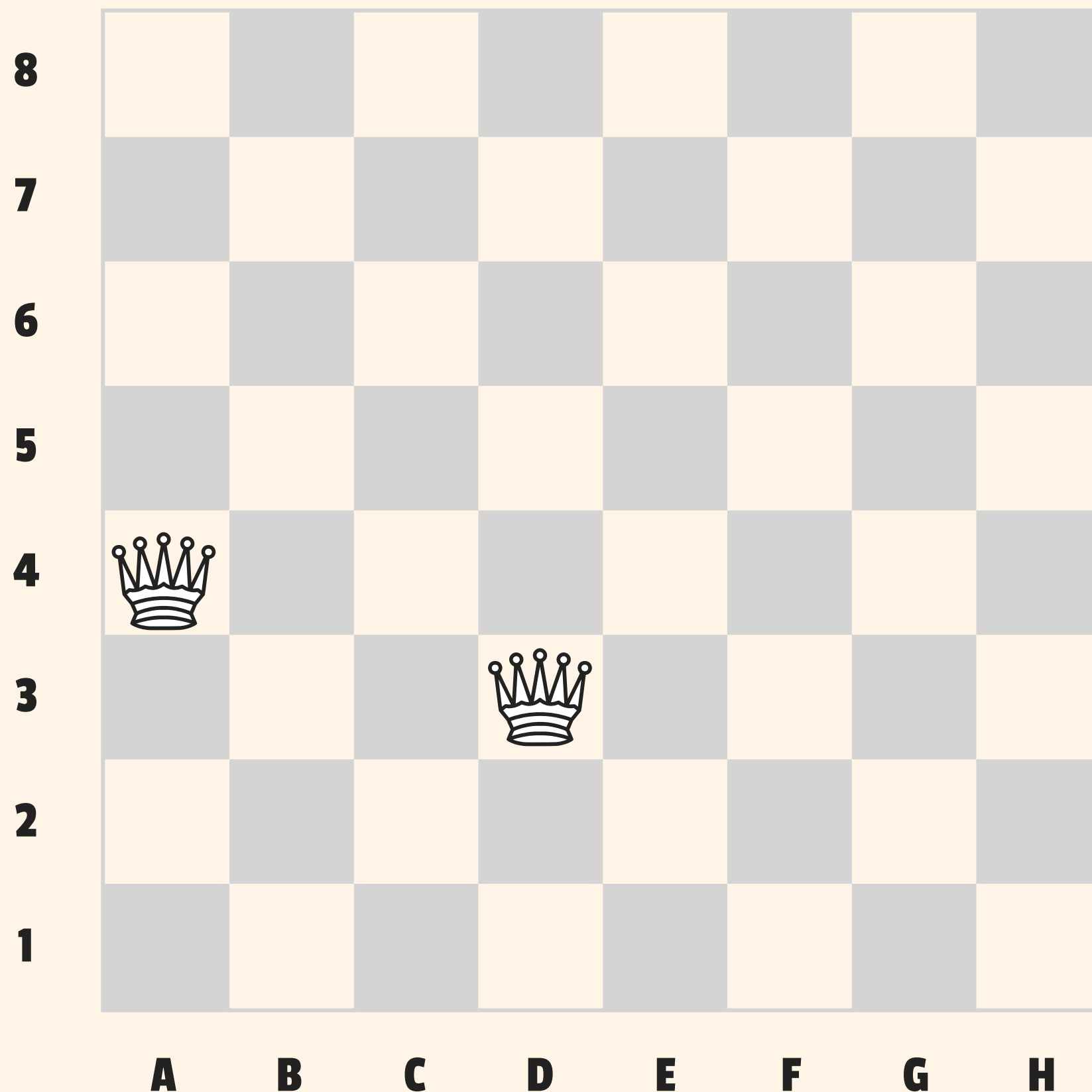
# 1 PROBLEMA

## **N-RAINHAS**

**Consiste em encontrar todas as combinações possíveis de  $N$  rainhas num tabuleiro de dimensão  $N$  por  $N$  tal que nenhuma das rainhas ataque qualquer outra**

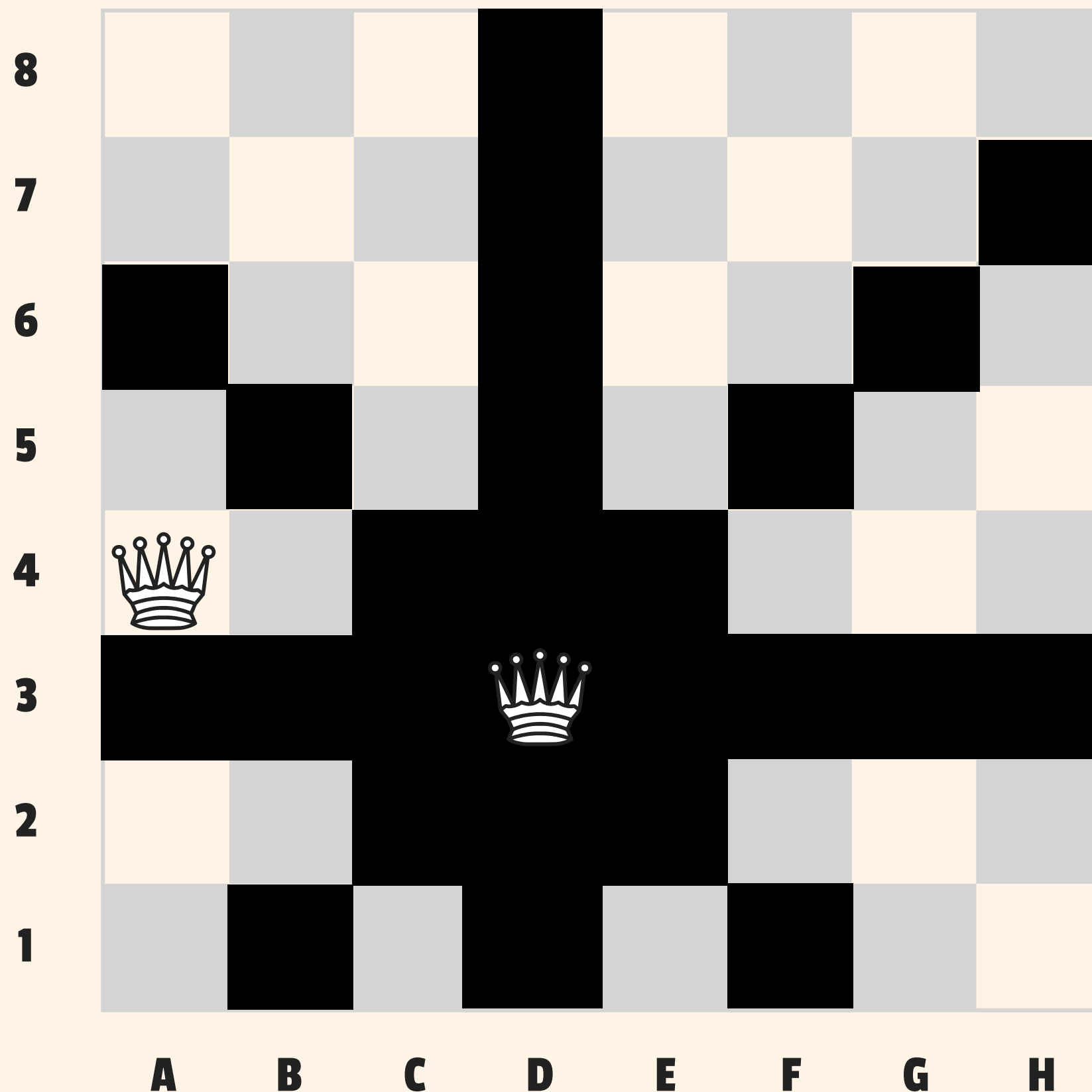
# COMO FUNCIONA N-RAINHAS

**Num tabuleiro de N por N**  
**Máximo de Rainhas sem que esteja nas**  
**diagonais/vertical/horizontal entre elas**



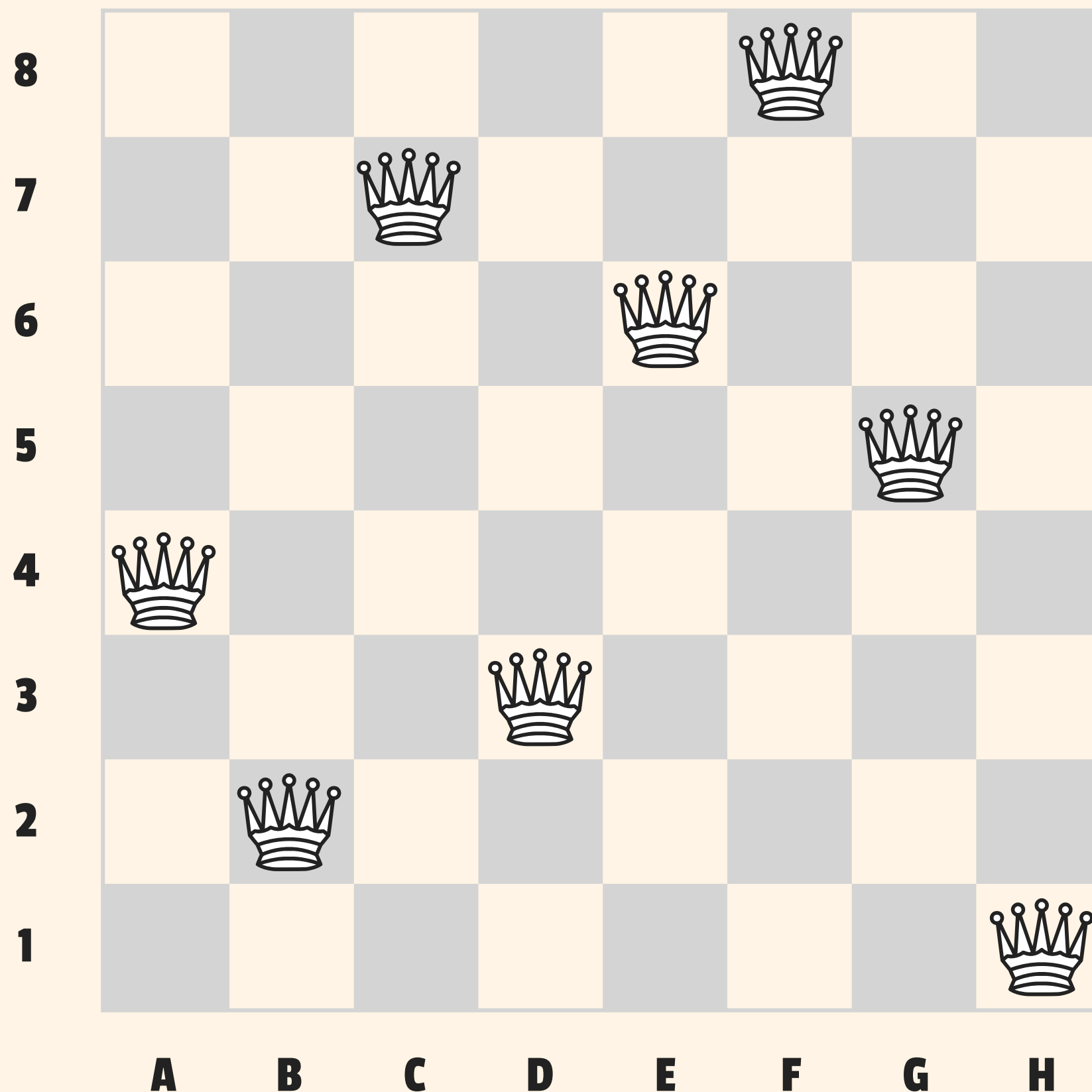
# COMO FUNCIONA N-RAINHAS

**Num tabuleiro de N por N**  
**Máximo de Rainhas sem que esteja nas**  
**diagonais/vertical/horizontal entre elas**



# COMO FUNCIONA N-RAINHAS

**Exemplo de uma Solução**





# 2 - TÉCNICAS

- **Algoritmo Genético**
- **Busca Tabu**



# ALGORITMO GENÉTICO

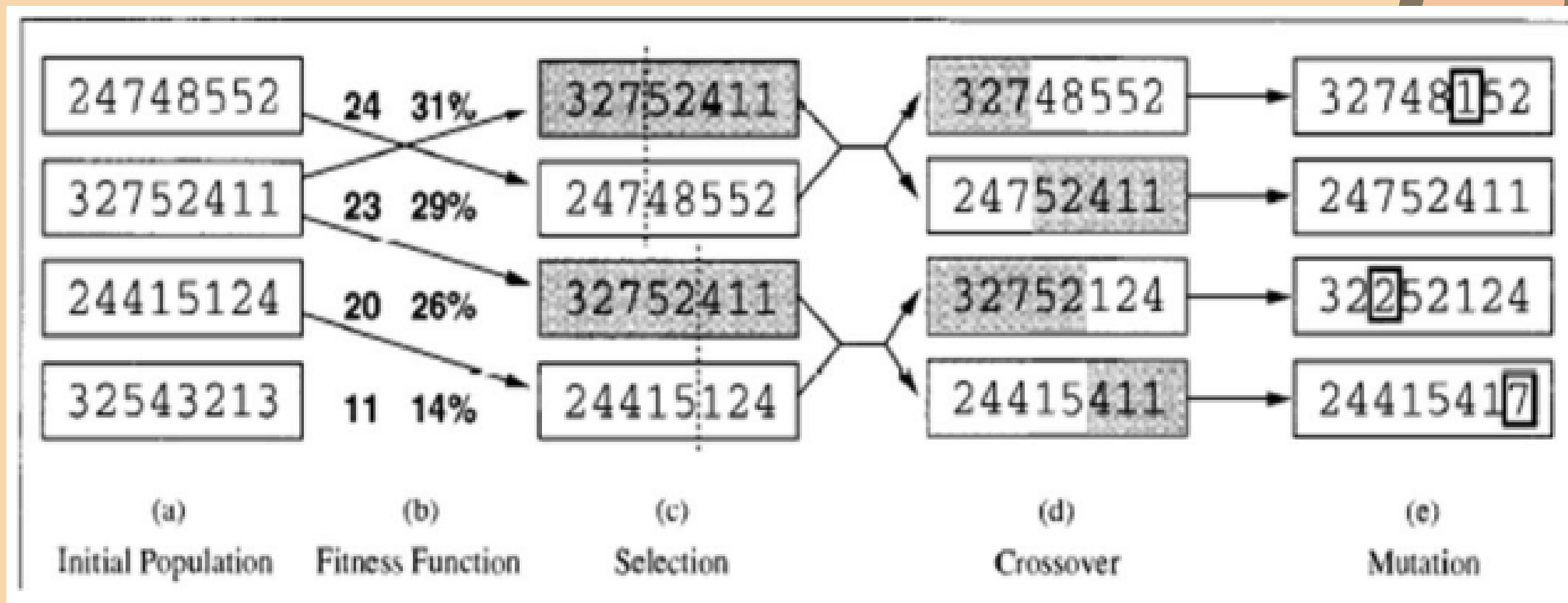
- **Introduzido por Holland em 1975**
- **k estados melhores do que os seus pais são gerados.**
  - **Um estado é gerado pela combinação de dois ou mais estados pais.**
  - **Com cada geração:**
    - **Pares aleatoriamente (+ aptos)**
    - **Crossover**
    - **Mutar alguns valores**





# ALGORITMO GENÉTICO

## Exemplo reprodução de estados



# ALGORITMO GENÉTICO

- **População: conjunto de  $k$  estados gerados aleatoriamente;**
- **Um estado é chamado de indivíduo/cromossomo**
  - **Estrutura de dados que representa uma possível solução.**
  - **Representado por uma cadeia de valores**



# ALGORITMO GENÉTICO

- **Função de aptidão (fitness)**
  - **Função para avaliar cada estado/indivíduo**
- **Em nossa implementação a aptidão é dada pela quantidade de rainhas que não se atacam.**
- **A aptidão máxima é dada pelo somatório de 1 até (numero de rainhas -1)**



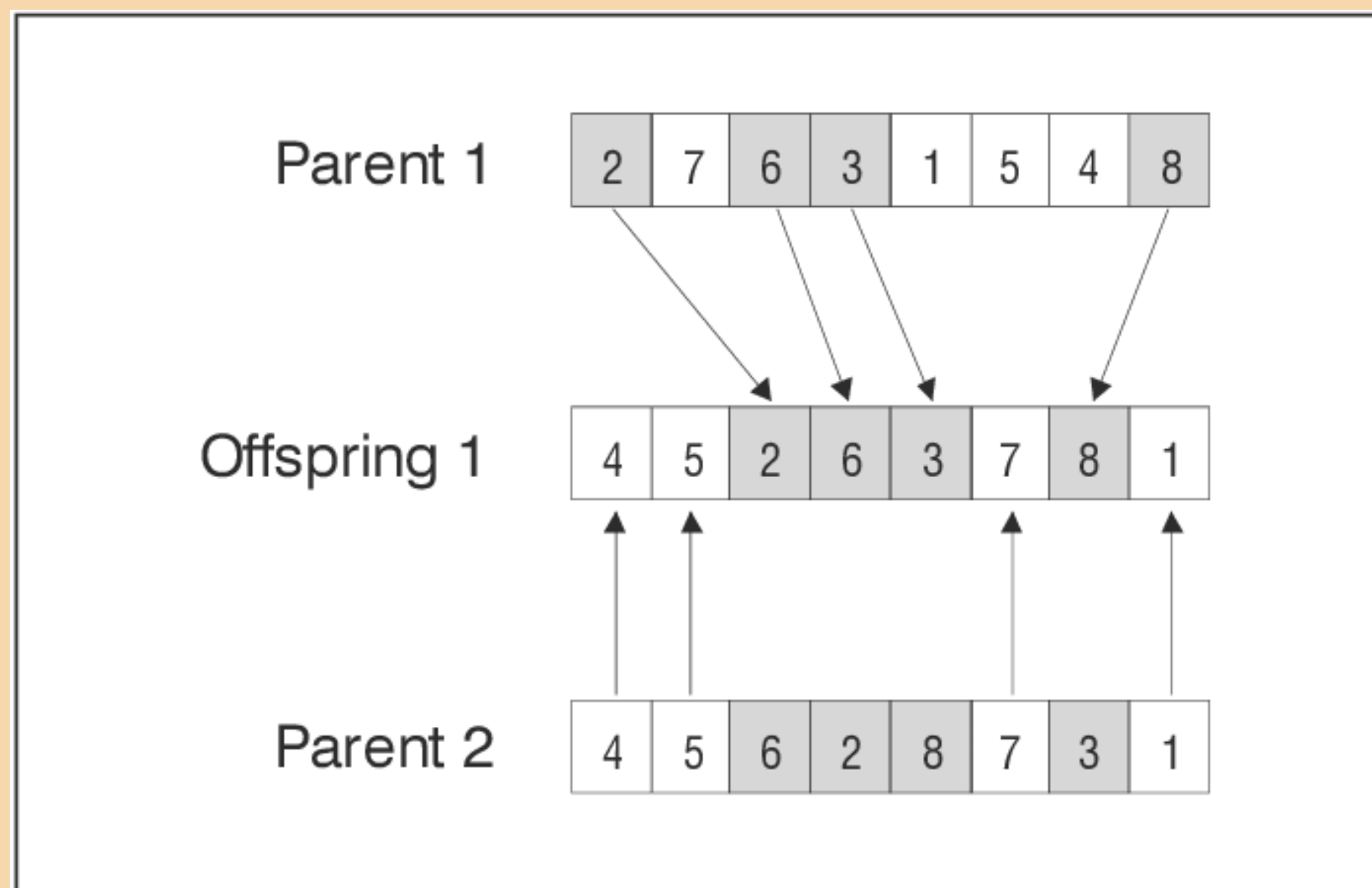
# ALGORITMO GENÉTICO

- **Reprodução dos indivíduos selecionados - Crossover**
  - **Cria novos indivíduos misturando características de dois ou mais indivíduos pais (crossover) - variação.**
- **Objetivo**
  - **Combinar o material genético dos mais adaptados**
  - **Explorar estados longe dos pais iniciais**
  - **Principais mecanismo de busca do AG**



# ALGORITMO GENÉTICO

## Exemplo de crossover - Order Based Crossover

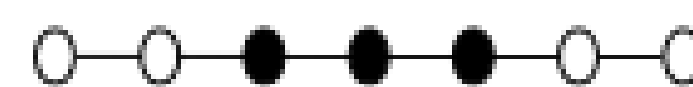
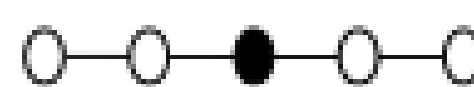


# ALGORITMO GENÉTICO

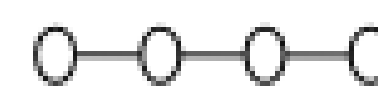
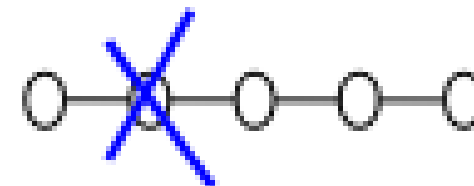
## Exemplo dos tipos de mutação

Estamos utilizando o Swap

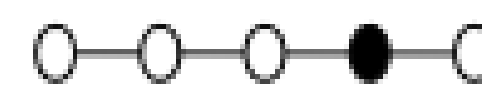
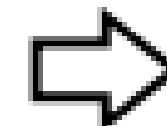
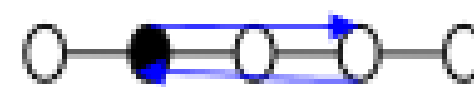
■ Gerativa



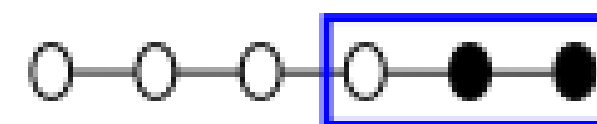
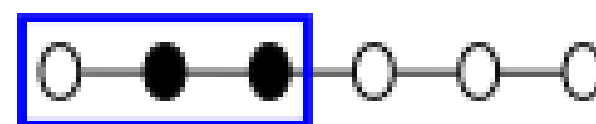
■ Destrutiva



■ Swap



■ Swap de sequência



# ALGORITMO GENÉTICO

## **Critério de parada**

- **Número de gerações**
- **Encontrou a solução (quando se sabe)**
- **Perda de diversidade**
- **Convergência**





# SEARCH TABU

- **Meta-heurística é uma estratégia geral que é usada para guiar e controlar heurísticas reais**
- **Search Tabu então é usada para otimizar os parâmetros do modelo escolhido**
- **integra estruturas de memória em estratégias de busca local. Como a pesquisa local tem muitas limitações, o Tabu Search foi projetado para combater muitos desses problemas.**



# SEARCH TABU

- **Ideia**
  - **Penalização de movimentos que levem a solução a espaços de busca visitados anteriormente**
  - **As vezes aceita determinísticas soluções q não melhoram para não ficar preso em mínimos locais**



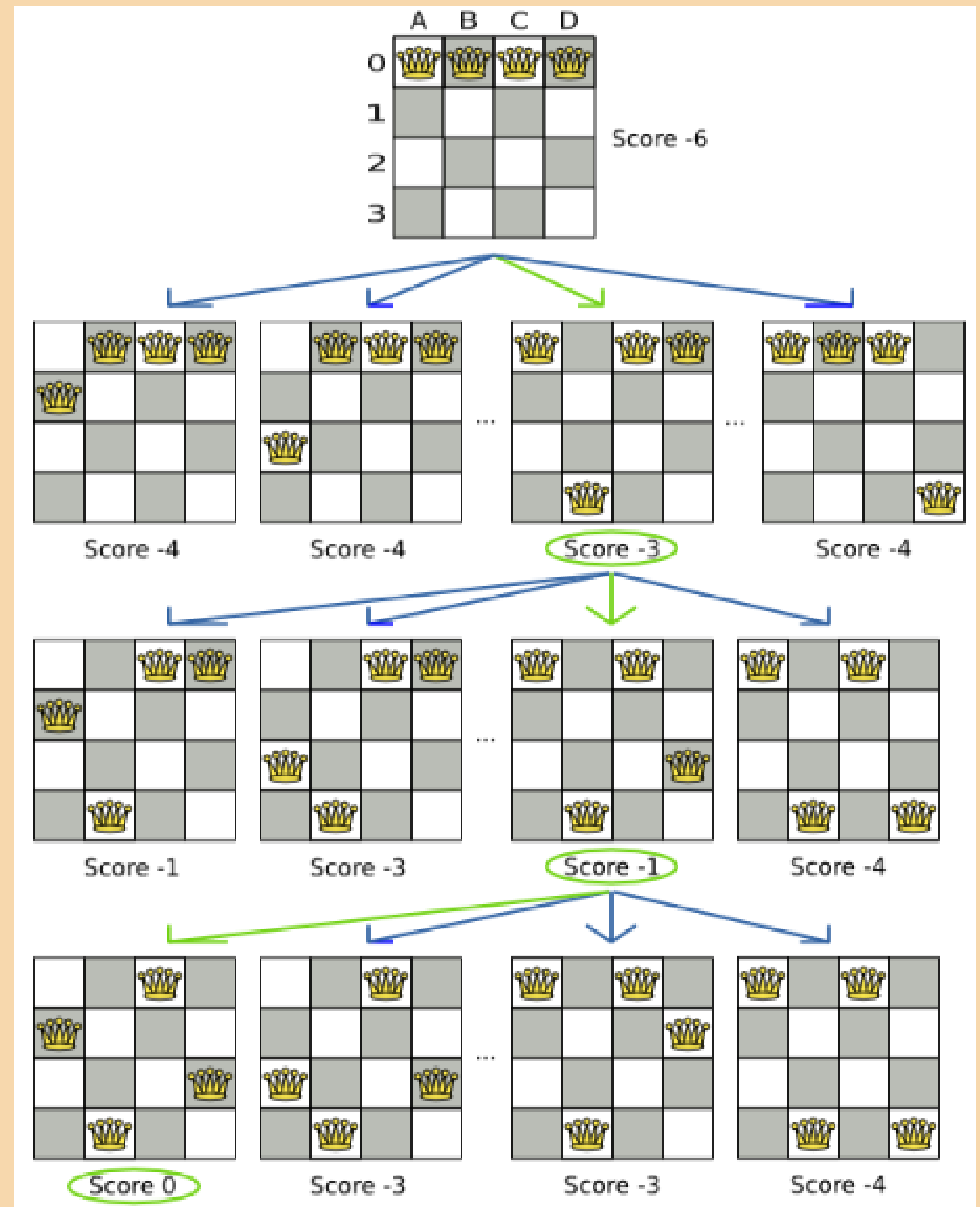
# SEARCH TABU

- **Memória de curto prazo**
  - Baseada na recência da ocorrência
  - Usada para evitar que o algoritmo de busca revise as soluções visitadas anteriormente
  - Usada para retornar a bons componentes a fim de localizar e intensificar uma busca
- **Memória de longo prazo**
  - Baseada na frequência de ocorrência
  - Diversificar a pesquisa evitando áreas exploradas



# SEARCH TABU

- Exemplo com n-rainhas 4x4



# 3 – EXPLICAÇÕES

## ALGORITMO GENÉTICO

**Linguagem = Python3**

**Complexidade do Problema**

- **Baixa (10 damas)**
- **Média (15 damas)**
- **Alta (20 damas)**

**Tamanho da População**

- **100**

**Probabilidade Mutação**

- **0.1**

**Max. Fitness**

- **Somatório de pares de rainhas que não se atacam**
- **Num. de rainha \* (Num. de rainha - 1) / 2**



# 3 – EXPLICAÇÕES

## ALGORITMO GENÉTICO

- Enquanto não chegar ao Fitness Máximo dos crossover dentre a população
- => Continua a busca com a próxima geração
  - Realiza o cruzamento
    - Tendo como escolha o fitness de cada cromossomo
    - Verifica a não repetição dos elementos
  - Adiciona na nova população
  - Fitness atual



# 3 – EXPLICAÇÕES

## SEARCH TABU

**Linguagem = Python3**

**Complexidade do Problema**

- **Baixa (12 damas)**
- **Média (18 damas)**
- **Alta (22 damas)**

**Prazo de Memória**

- **100**





# 3 – EXPLICAÇÕES

## SEARCH TABU

- **Enquanto houve conflitos de damas, continue a busca**
  - **Selecione aleatoriamente uma dama**
  - **Considere a melhor e realiza a troca de posição**
    - **Verifica a quantidade de conflitos**
    - **Se melhor, troca, se não mantenha a anterior**
  - **Verifique os vizinhos**
    - **Repita os trocas de posição**
    - **Verifica se conflito diminuiu**
  - **Selecione a opção de menor conflito e sega com a iteração**



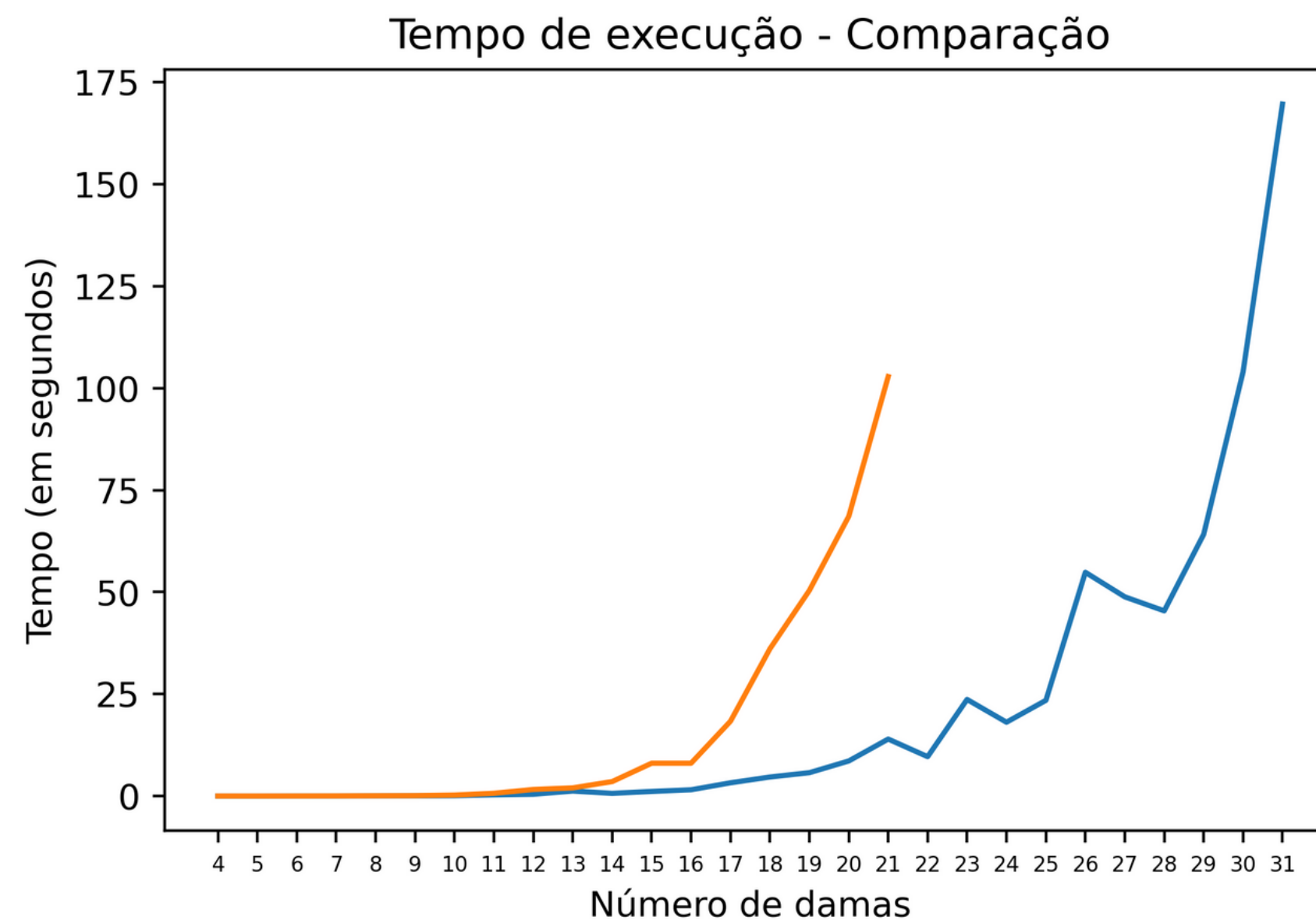
# 4 - RESULTADOS

## Comparação entre

- **Algoritmo Genético (laranja)**
- **Busca tabu (azul)**

- **Sendo:**

- **Tempo em segundos x**
- **Número de damas**



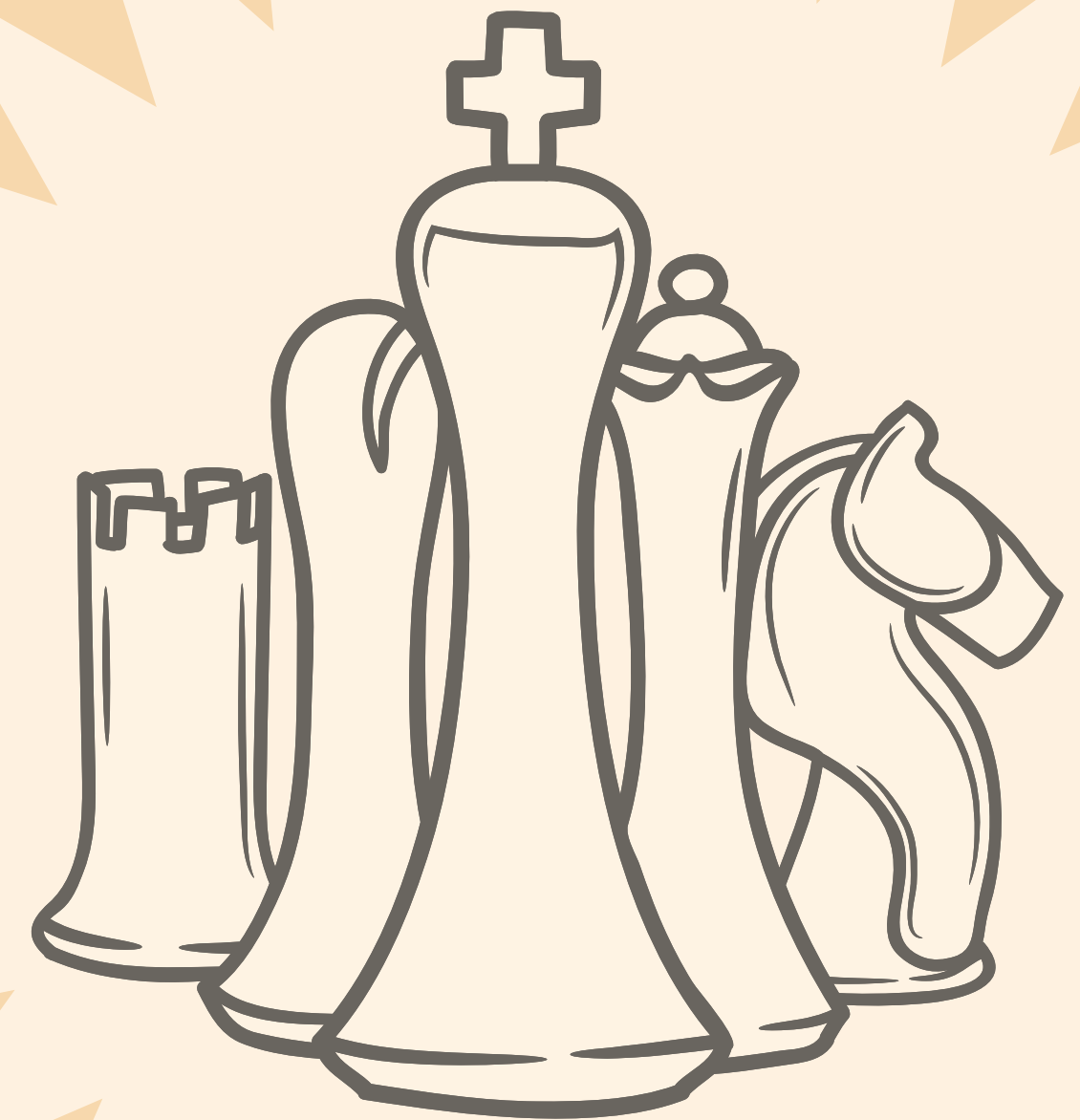
# CONCLUSÕES

- **Ambas soluções chegaram no resultado, utilizando heurísticas diferentes.**
- **Tabu foi mais eficiente, tendo encontrado soluções ótimas em tempo menor**
- **Já no genético, poderia ser mais otimizado porém fora suficiente a medida apesar de demorar mais a medida que se tem maiores soluções ótimas**
- **Contribuição na ampliação do conhecimento sobre problemas de busca**



# REFERENCIAS

- **Slides Aula**
- **[www.cin.ufpe.br/~if684/aulas/algoritmosgeneticostbl.ppt](http://www.cin.ufpe.br/~if684/aulas/algoritmosgeneticostbl.ppt)**
- **RUSSEL, Stuart Jonathan; NORVIG, Peter. Inteligência Artificial. Tradução por Regina Célia Simille, Artificial intelligence, 3rd ed. Rio de Janeiro: Elsevier, 2013.**
- **<https://towardsdatascience.com/optimization-techniques-tabu-search-36f197ef8e25>**
- **<https://docs.optaplanner.org/7.0.0.Beta1/optaplanner-docs/html/ch10.html>**



:)

GABRIEL T.  
SERGIO A.

**OBRIGADO**

