

Alunos: Bruno Campos, Elisa de Mattos Rosá e Filipe Voges

# Trabalho 1 Problema das N-Rainhas

**Elaborar um relatório detalhado explicando brevemente as principais soluções encontradas na pesquisa, apresentando em detalhes os dois algoritmos escolhidos juntamente com os programas executáveis das duas soluções implementadas, fazendo uma comparação entre os resultados obtidos com cada uma delas.**

## **Força bruta:**

A primeira solução encontrada foi a de força bruta, contudo, analisando melhor, foi possível perceber que a abordagem de força bruta não é prática pois sempre será feita uma combinação do número de posições com o número de rainhas.

## **Recursão com Backtracking:**

Backtracking é um tipo de algoritmo parecido com força bruta só que aperfeiçoado. Primeiro é gerado todas as maneiras possíveis de inserir N rainhas nas N linhas.

Para gerar as posições válidas para a rainha  $N + 1$ , colocamos uma rainha em todas as colunas da linha  $N + 1$  e rejeitamos os estados inválidos. Se um local válido é encontrado, a função `put_rainha()` chama a si mesma tentando colocar uma rainha nesta posição. Para conseguir inserir uma rainha primeiro é chamado a função `backtracking` que verifica se uma determinada posição pode ser atacada por alguma rainha inserida anteriormente. Fazemos os passos acima até as oito rainhas serem colocadas no tabuleiro.

Por fim, imprimimos todas as soluções possíveis com a função `imprime_tabuleiro()`.  
OBS: foi implementado em Python 3.6

[https://github.com/brunocampos01/inteligencia-artificial/tree/master/trabalho01-n\\_queens](https://github.com/brunocampos01/inteligencia-artificial/tree/master/trabalho01-n_queens)

## **Hill climbing:**

Este algoritmo de busca local é simplesmente um laço repetitivo que se move de forma contínua no valor crescente. Em cada passo o nó corrente é substituído pelo melhor vizinho.

OBS: foi implementado em PHP

<https://github.com/FilipeVoges/n-rainhas/blob/master/NQueens1.php>

## **Bibliografia:**

<https://www.geeksforgeeks.org/n-queen-problem-backtracking-3/> acesso 01/09/2018

<https://www.coursera.org/lecture/what-is-a-proof/n-queens-backtracking-example-optional-YiULQ> acesso 01/09/2018

Russell S., Norvig Peter; INTELIGÊNCIA ARTIFICIAL; 2 ed autor Russell e Norvig, pg 110