

Atividade 5: oito rainhas (conjuntos independentes)

Gustavo Henriques da Cunha

Ciência da Computação

Grafos

UFSJ

09/12/2023

1 Introdução

Esta atividade têm como motivação a prática dos algoritmos vistos em aulas, buscando achar uma solução para o famoso problema das oito rainhas, utilizando uma abordagem modelada com grafos, utilizando conjuntos independentes.

2 Instruções de execução

Para executar a solução é necessário o arquivo 'Tabuleiro_com_incompatibilidades.gml' estar na mesma pasta do arquivo. Como resultado, será imprimido no terminal as posições das rainhas.

3 Solução

Para resolvermos o problema, foi utilizada a linguagem *Python*, com a biblioteca *NetworkX*, devido a facilidade de lidar com as estruturas de dados de grafos usando a biblioteca.

Como dito anteriormente, foi utilizado de uma abordagem com grafos, mais especificamente do conjunto independente de vértices, para resolver o problema.

Assim, foi abordada uma heurística simples para solucionar o problema. Ela consiste em selecionar um vértice aleatório no grafo, adicioná-lo ao conjunto, e depois remover todos seus vizinhos e ele mesmo do grafo, repetindo o processo até o grafo estar vazio. Entretanto, isso não irá garantir uma solução do problema, por isso, repetimos o algoritmo até encontrar uma solução com 8 vértices, que seria a posição das 8 rainhas.

3.1 Algoritmo

Segue o código em Python da solução:

```
1 def independent_set (G: nx.Graph) -> list:
2     i_set = []
3     while G.number_of_nodes():
4         node = choice(list(G.nodes()))
5         i_set.append(node)
6         for n in list(G.neighbors(node)):
7             G.remove_node(n)
8         G.remove_node(node)
9     return i_set
```

Chamamos a função até a lista de retorno ter tamanho 8.

4 Resultados

Executando o algoritmo várias vezes, encontramos diferentes respostas ao problema, como as duas abaixo:

```
050024_atividade5/rainhas.py
15
49
19
30
32
60
45
2
(base) gustavo@gustavo-Aspire
24_atividade5/rainhas.py
58
31
5
11
36
17
48
46
```

Figura 1: Resultados 1 e 2

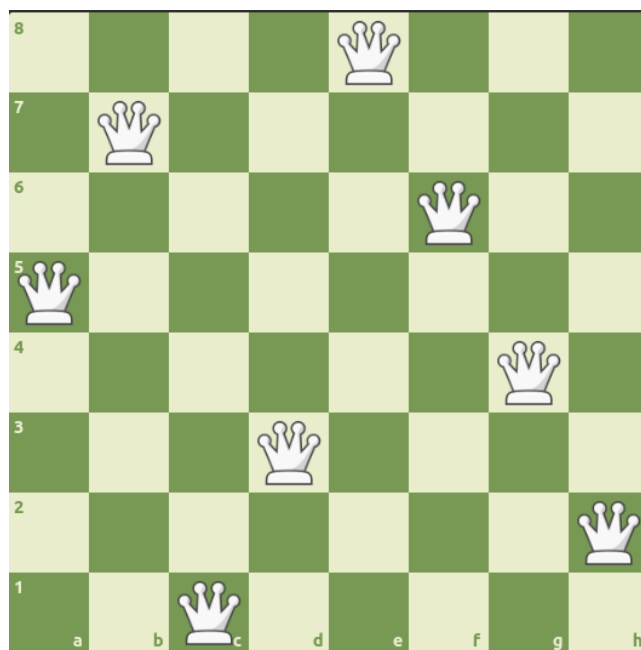


Figura 2: Vizualização do resultado 1

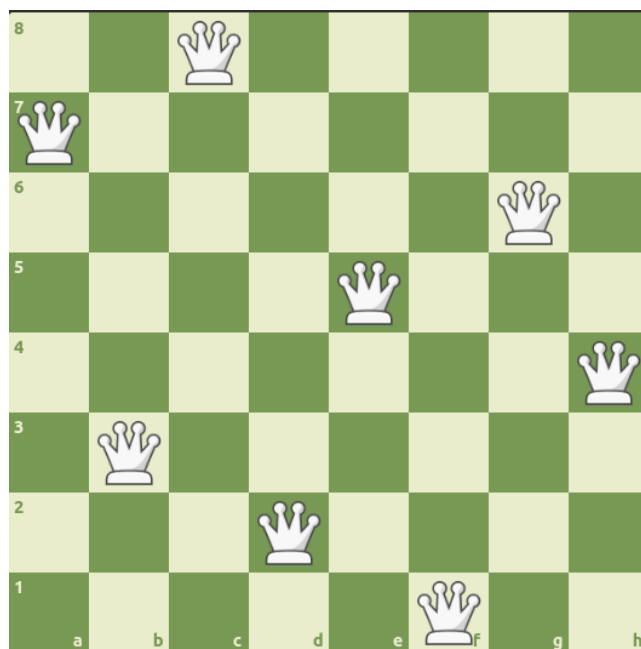


Figura 3: Vizualização do resultado 2