

Samuel Duarte Fernandes Lima – 2216976

## 1. INTRODUÇÃO

Demonstração dos algoritmos aplicados no jogo 8 Puzzle: Breadth-first Search (BFS), Depth-first Search (DFS), Iterative Depth-first Search (IDFS), Greedy e Hill Climbing. A matriz objetivo é embaralhada com um número específico de movimentos, neste caso apenas 50, gerando uma configuração aleatória inicial. Essa abordagem visa facilitar a observação do comportamento dos algoritmos durante o jogo, evitando que possam levar a ciclos infinitos ou soluções inviáveis.

Sendo as matrizes objetivos:

1	2	3
8		4
7	6	5

1	2	3	4
12	13	14	5
11		15	6
10	9	8	7

## 2. HEURÍSTICAS

As heurísticas escolhidas foram: A primeira, baseada no número de peças que estão na mesma posição da matriz objetivo, avaliando quantas peças já estão corretamente posicionadas; A segunda é a de Manhattan, que calcula a soma das distâncias horizontais e verticais de cada peça fora de lugar até sua posição correta no tabuleiro.

## 3. COMPARATIVO DE RESULTADOS

Configuração do computador utilizado para a pesquisa: Processador Ryzen 7 5700x, 16 GB de RAM, Placa de vídeo RTX 2060.

Entre todos os métodos de busca cega testados, a busca em largura (BFS) destacou-se como a mais eficiente tanto em termos de tempo quanto de uso de memória. Para o quebra-cabeça de 15 peças, a BFS apresentou um consumo máximo de memória de 300 MB e um tempo de execução de aproximadamente 10 segundos. No quebra-cabeça de 8 peças, o consumo máximo de memória foi de 134 MB, com um tempo de execução de 6 segundos. Em alguns casos raros, tanto para o quebra-cabeça de 8 peças quanto para o de 15 peças, o tempo e o consumo de memória podem mais do que dobrar. Por outro lado, os

valores mínimos observados foram de 32 MB de memória e menos de 0,1 segundos para ambos os quebra-cabeças.

Em contraste, o método de busca em profundidade (DFS) foi o mais ineficaz, apresentando casos em que utilizou o máximo de memória, entre 8 e 12 GB, disponível no computador e levou até 90 segundos para encontrar a solução, tanto para o quebra-cabeça de 8 peças quanto para o de 15 peças, fora que muitos dos casos acabam tendendo ao infinito. Porém nos casos ótimos utilizava 32 MB e tempo inferior a 0.1 segundos.

A busca por profundidade iterativa (IDFS) apresentou um desempenho muito semelhante ao da BFS em termos de uso de memória, com o consumo máximo de memória sendo comparável. No entanto, o tempo de execução da IDFS foi geralmente maior, refletindo o custo adicional de realizar múltiplas iterações de profundidade. Embora o benchmark da IDFS seja bastante próximo ao da BFS, o aumento no tempo de execução pode ser significativo em alguns casos, enquanto a eficiência no uso de memória permanece favorável.

Em relação aos métodos heurísticos, a busca gulosa (Greedy) apresentou um desempenho de tempo comparável ao da busca em largura (BFS), com variações principalmente no tempo de execução. No entanto, o algoritmo de Hill Climbing não conseguiu encontrar uma solução para os quebra-cabeças devido à sua tendência a ficar preso em máximos locais, sem considerar uma estratégia global de solução. Além disso, a heurística baseada na contagem de posições corretas demonstrou um desempenho superior em termos de tempo quando comparada à heurística de distância de Manhattan.

#### **4. CONCLUSÃO**

Portanto, mesmo que os métodos heurísticos possam parecer mais sofisticados, a BFS oferece um excelente custo-benefício ao equilibrar a precisão da solução com a eficiência de tempo e memória, tornando-a a escolha preferencial para resolver quebra-cabeças de 8 e 15 peças.