Resolução do Slide Puzzle com Algoritmos de Busca

Este relatório apresenta a implementação e análise comparativa de três algoritmos clássicos de busca. O objetivo é avaliar a eficiência de cada algoritmo quanto ao tempo de execução, número de nós visitados e qualidade da solução (número de movimentos).

Os algoritmos avaliados:

Busca em Largura (Breadth-First Search - BFS)
Busca em Profundidade (Depth-First Search - DFS)
Busca A* com duas heurísticas: Distância Manhattan e Misplaced

Link para o codigo: https://github.com/acmachado48/IA/tree/main/PUZZLE

1. Busca em Largura (BFS)

A Busca em Largura explora uniformemente todos os nós em um mesmo nível de profundidade antes de explorar nós mais profundos. É implementada com uma estrutura de fila (FIFO).

Características:
Garante encontrar a solução ótima (menor número de movimentos).
Consome grande quantidade de memória em problemas complexos.

2. Busca em Profundidade (DFS)

A Busca em Profundidade explora um caminho até o fim antes de retroceder, utilizando uma pilha (LIFO). Pode ser implementada iterativamente ou de forma recursiva.

Características:

Consome menos memória.
Pode explorar caminhos muito longos e irrelevantes.
Não garante a solução ótima.

3. Busca A* (A-star)

A busca A* é um algoritmo heurístico que combina o custo real do caminho até o nó g(n) com uma estimativa do custo restante até o objetivo h(n), formando a função:

$$f(n) = g(n) + h(n)$$

Duas heurísticas foram utilizadas:

a) Peças Fora do Lugar (Misplaced Tiles)
Conta o número de peças fora da posição correta.

b) Distância Manhattan

Soma das distâncias horizontais e verticais de cada peça até sua posição final.Mais precisa, admissível e com melhor orientação da busca.

Resultados Experimentais

Todos os algoritmos resolveram a mesma configuração inicial do puzzle 3x3 (8-puzzle). Os resultados foram medidos em tempo de execução (segundos), quantidade de nós visitados e número total de movimentos até a solução.

Algoritmo	Tempo (s)	Nós Visitados	Movimentos
A* (Misplaced)	0.01	3555	20
A* (Manhattan)	0.01	204	22
DFS	0.30	180158	1596
BFS (Falha)	0.00	0	0
BFS (Completa)	0.36	85765	22

Análise Comparativa

A* (Manhattan) vs A* (Misplaced)

Eficiência: A* com Manhattan visitou 204 nós, contra 3555 da heurística Misplaced.
Qualidade da Solução: Manhattan resultou em 22 movimentos, Misplaced em 20.
Conclusão: A heurística de Manhattan é mais eficiente computacionalmente, com ligeira perda na qualidade da solução.

A* vs DFS

DFS foi drasticamente ineficiente, com 180 mil nós visitados e caminho de 1596 movimentos.
A* (Manhattan) foi 900x mais eficiente em termos de nós explorados.

A* vs BFS

Ambos encontraram soluções com 22 movimentos.
BFS visitou 85765 nós, enquanto A* (Manhattan) apenas 204.
A* foi 400x mais eficiente na exploração.

Conclusão

A análise demonstra que:

O algoritmo A* com heurística de Manhattan é o mais eficiente e eficaz na resolução do puzzle.
A heurística de peças fora do lugar é admissível, mas inferior à Manhattan em desempenho.
A busca em profundidade é inadequada para esse tipo de problema, tanto em tempo quanto em qualidade de solução.
A busca em largura, embora ótima, é menos eficiente que A*