

Relatório do Código do Puzzle 8 - Busca A* e BFS

1. Introdução

Este relatório apresenta a implementação de um solucionador para o quebra-cabeça 'Puzzle 8', usando dois algoritmos clássicos de busca: A* (A-estrela) e BFS (Busca em Largura). O Puzzle 8 é um problema clássico de busca em inteligência artificial, onde o objetivo é mover as peças numeradas em um tabuleiro 3x3 até uma configuração final ordenada, com o espaço vazio (representado por 0) na última posição.

2. Descrição Geral do Código

O código consiste na classe `Puzzle8`, que encapsula o estado inicial, estado objetivo, as regras de movimentação e os algoritmos de busca. A solução utiliza uma representação imutável dos estados por meio de tuplas para facilitar a manipulação e verificação de estados visitados.

3. Partes Principais do Código

3.1 Inicialização da Classe `Puzzle8`

No método `__init__`, são definidos:

- O estado inicial fornecido pelo usuário.
- O estado objetivo predefinido.
- Um dicionário de movimentos possíveis para o espaço vazio, que determina para onde ele pode se mover a partir de cada posição.
- Um mapeamento das posições dos números no estado objetivo para acelerar o cálculo heurístico.

3.2 Heurística (função `heuristic`)

A função `heuristic` calcula a soma das distâncias de Manhattan das peças do estado atual até suas posições corretas no estado objetivo. Esta heurística é admissível, ou seja, ela nunca superestima o custo para alcançar o objetivo, o que garante a eficiência e corretude do algoritmo A*.

3.3 Algoritmo A* (função astar)

O algoritmo A* utiliza uma fila de prioridade para explorar os estados com menor custo estimado primeiro. Cada estado é avaliado pelo custo atual (número de movimentos feitos) somado à heurística. Estados já visitados com custos maiores são descartados para otimizar a busca. O algoritmo termina quando o estado objetivo é encontrado, retornando o caminho e o número de estados testados.

3.4 Algoritmo BFS (função bfs)

A busca em largura (BFS) é implementada como método alternativo. Ela explora todos os estados de forma sequencial e não usa heurística. Embora garanta encontrar a solução mínima em número de movimentos, é menos eficiente para problemas como o Puzzle 8 devido ao grande espaço de estados.

3.5 Função Principal (main)

A função main realiza a leitura da entrada do usuário, valida os dados para garantir que corresponda ao formato esperado e inicializa o objeto Puzzle8. Em seguida, chama o algoritmo A* para resolver o puzzle, exibindo a solução encontrada e o número de estados testados, ou informa se não há solução.

4. Conclusão

Este código fornece uma solução prática para o Puzzle 8 utilizando uma abordagem eficiente com o algoritmo A*, aliado a uma heurística admissível. A implementação é modular, facilitando testes e extensões, como a inclusão de outros algoritmos de busca ou melhorias na heurística. Além disso, a função BFS serve como comparação didática para entender as vantagens do A*.