Universidade Federal de Minas Gerais **Inteligência Artificial**
Instituto de Ciências Exatas **1º Semestre de 2019**
Departamento de Ciência da Computação **Prof. Luiz Chaimowicz**

Trabalho Prático 1

8-Puzzle

Data de entrega: 30/04

O objetivo do trabalho consiste em implementar e comparar os diferentes métodos de busca apresentados durante o curso, aplicando-os a um toy problem. O quebra-cabeça das oito peças (8-Puzzle) é composto por uma moldura 3x3 contendo um conjunto de peças numeradas de 1 a 8 e um espaço vazio. O propósito do jogo resume-se a posicionar as peças em uma determinada ordem (Figura 1) apenas deslizando-as pela moldura. Sabe-se que o problema de encontrar uma solução com o menor número de passos no caso geral, denominado N-Puzzle, é NP-Difícil [1].

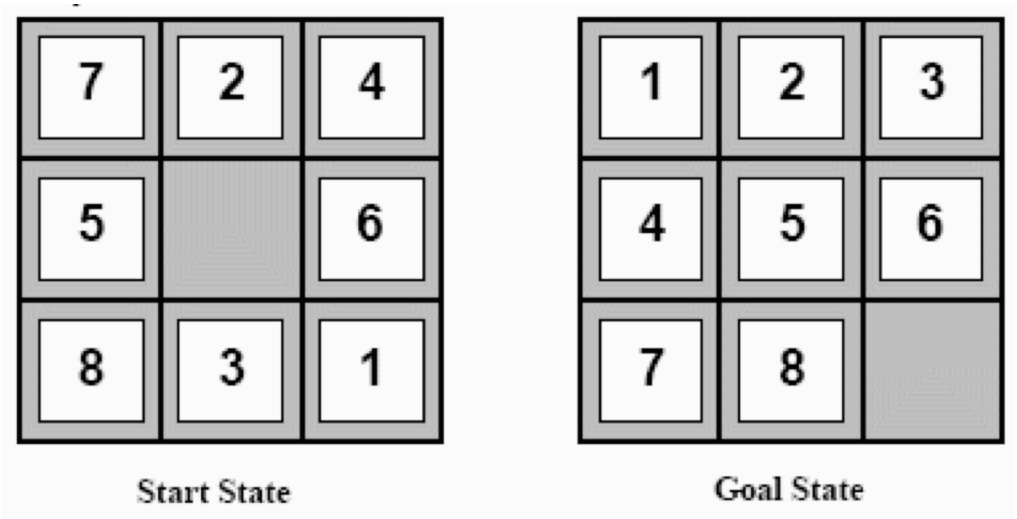

Figura 1. *Estados inicial e final para o 8-Puzzle.*

Vocês devem implementar, em uma das seguintes linguagens (C, C++, Java ou Python), as estruturas de dados, heurísticas e algoritmos necessários para resolver esse quebra-cabeça. Particularmente, os seguintes algoritmos de busca devem ser implementados e comparados:

Busca sem informação	*Breadth-first search* *Iterative deepening search* *Uniform-cost search*
Busca com informação	*A* search* *Greedy best-first search*
Busca local	*Hill Climbing*, permitindo movimentos laterais.

Com relação aos algoritmos A* e Greedy best-first, vocês devem utilizar **duas heurísticas distintas**. Já o Hill Climbing vai ter que ser ligeiramente alterado para salvar o caminho encontrado e permitir movimentos "laterais" no espaço de estados até um limite *k*.

O que deve ser entregue:

- Códigos fonte dos algoritmos desenvolvidos
- Um arquivo readme.txt com informações sobre como compilar e executar os seus programas
- Documentação contendo uma **descrição sucinta** das estruturas de dados, heurísticas e algoritmos empregados para modelar e resolver o problema, como também uma **discussão dos resultados obtidos.**

Mais especificamente, esse documento deve incluir:

- Modelagem dos componentes básicos da busca (estado, função sucessora, etc);
- Breve descrição das principais diferenças entre os algoritmos;
- Especificação das heurísticas utilizadas. Elas são admissíveis? Por quê?;
- Exemplos de soluções encontradas pelos algoritmos. Particularmente, apresente as soluções para a última instância (Solução: 31) do arquivo *npuzzle.pdf*.
- Análise quantitativa comparando os algoritmos com relação ao número de estados expandidos, número de passos da solução e tempo de execução. **Apresente tabelas e/ou gráficos comparativos.**
- Discussão dos resultados obtidos.

Ponto Extra: Explore a versão NP-Difícil do problema. Altere a sua estrutura de dados para suportar o N-Puzzle e analise o comportamento do algoritmo A* para diferentes valores de N.

Bom trabalho!

[1] Daniel Ratner, Manfred K. Warmuth. *Finding a Shortest Solution for the N × N Extension of the 15-PUZZLE Is Intractable*. National Conference on Artificial Intelligence, 1986.