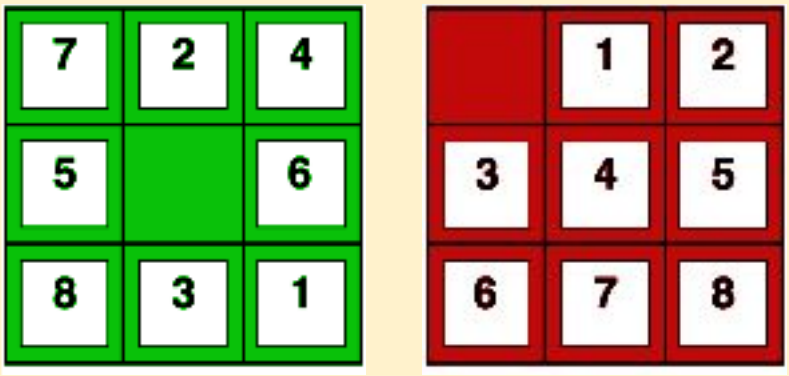
Quebra-cabeça 3x3

Objetivo

O objetivo desta atividade é a prática na construção de uma solução para problemas com o emprego da Busca em Espaço de Estados com informação. Os alunos devem construir um programa que implemente o problema do quebra-cabeça de blocos deslizantes 3x3, e uma função de busca baseada em alguma heurística. Então, descrever e avaliar os resultados.



Objetivos Específicos

1. Construir uma *implementação do quebra-cabeça de tamanho 3x3*, composto por números em ordem crescente (solução) e um espaço em branco.
2. Elaborar uma *função sucessora* que a partir de um determinado estado do problema (sequência de peças do quebra-cabeça) retorne os estados adjacentes do espaço de estados.
3. Elaborar uma *função busca* que, a partir de um estado inicial do problema, constrói o espaço de estados utilizando:
    a. *algoritmo de busca sem informação*, executando a busca com um algoritmo de busca em grafo (largura ou profundidade).
    b. *algoritmo de busca com informação*, executando a busca com um algoritmo A*.
4. *Coletar informações sobre o desempenho* da função de busca, como por exemplo:
    a. tempo de execução da função de busca;
    b. quantidade estimada de memória utilizada na busca da solução.

Entrega

O trabalho deve ser **desenvolvido individualmente** e entregue em duas partes: *código desenvolvido* e *resultados*.

1. *Código*:
    a. o código deve ser desenvolvido, testado e entregue pronto para execução. Recursos adicionais para execução devem ser incluídos para que não seja necessária instalação.
    b. O algoritmo de busca deve implementar especificamente o pseudocódigo de Luger Inteligência Artificial (Luger, 2013), seção 4.2.
    c. Entregas parciais deverão ser desenvolvidas e entregues de acordo com cronograma da atividade.
2. Resultados: o software deve apresentar na execução as seguintes informações:
    a. O estado inicial (tabuleiro).
    b. O espaço de estados construído durante a busca.
    c. A solução encontrada (caminho no espaço de estados).

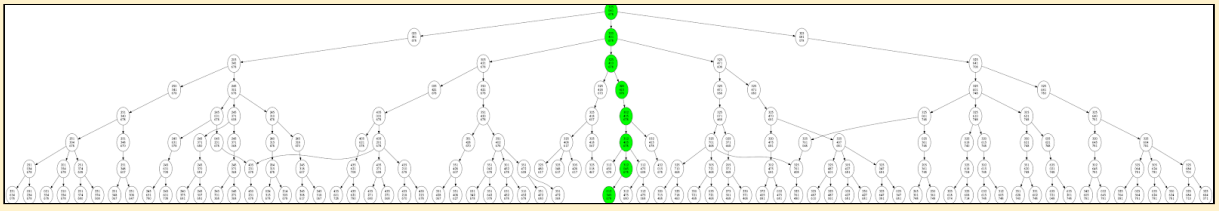
d. Medidas de desempenho (tempo de execução da busca e memória utilizada).
e. Para uma apresentação dos resultados:
    i. Fazer um vídeo com uma **demonstração do software**, sua **implementação** e **testes**.
    ii. O vídeo deve ter **no máximo 10 minutos** de duração (podem ser vários arquivos, organizados, num total de 10 minutos) e ser **enviado em conjunto com o código via Blackboard** (através de um link de Youtube ou arquivo de vídeo).
    iii. **Sugestão**: **para gravar a tela** é possível utilizar a Barra Gamer do Windows 10 ou o software ShareX, por exemplo.

Avaliação

A avaliação do trabalho será feita ao longo desenvolvimento e com base no material da entrega. A nota final será constituída pelo atendimento aos requisitos solicitados na entrega: código (70% da nota) e apresentação dos resultados (30% da nota).

Dicas e observações

- construa o código de forma gradativa e priorize:
    a. a representação do problema e sua função sucessora;
    b. implemente a função de busca sem informação em largura, por exemplo, para então implementar uma busca com informação;
    c. consulte o livro Luger Inteligência Artificial (Luger, 2013), seção 4.2, para obter dicas importantes da solução e implementação.
    d. Não crie estados iniciais aleatórios, pois eles podem não ter solução.
- Exemplo de solução com Busca em Largura:
    a. tabuleiro [[3,2,5],[0,4,1],[6,7,8]]



    b. tabuleiro [[0,4,1],[3,2,5],[6,7,8]]