

Primeiro Trabalho de Sistemas Inteligentes

Entrega: 23/03/2015

6 de Março de 2015

Este trabalho vale nota e será utilizado para calcular a média final junto com as notas das provas e dos outros trabalhos. Para saber o método e critério de avaliação, por favor consulte a ficha da unidade curricular na página do sigarra.

O jogo dos 8 é representado por uma matriz 3x3 onde há 8 células numeradas e uma célula em branco. Variações deste jogo podem conter parte de uma imagem em cada célula. O problema consiste em partir de uma configuração inicial embaralhada das células e chegar a uma configuração final com uma ordenação determinada de algarismos (no caso da matriz de números) ou de imagens (no caso da matriz onde as células representam partes de uma imagem). Os movimentos/operadores possíveis para se chegar de uma configuração a outra são: 1) mover a célula em branco para cima, 2) mover a célula em branco para baixo, 3) mover a célula em branco para a direita e 4) mover a célula em branco para a esquerda.

Dada a descrição do problema acima, implemente uma solução usando:

- as estratégias de busca não informadas **busca em profundidade**, **busca em largura** e **busca em profundidade iterativa** e aplique a este problema.
- as estratégias de busca informadas **busca gulosa** e **busca A*** e aplique a este problema.

A implementação pode ser em qualquer linguagem.

Para cada estratégia, analise:

- tempo para chegar a uma solução,
- quantidade de espaço gasto (número de nós gerados e armazenados),
- completude (o algoritmo consegue encontrar uma solução?) e
- otimalidade (o algoritmo consegue encontrar a solução no menor número de passos ou com menor custo em tempo hábil e utilizando pouca memória? Qual é a profundidade da solução?)

Utilize as seguintes configurações iniciais:

3	4	2
5	1	7
6		8

6	2	7
5		3
8	1	4

A primeira configuração tem solução ótima em profundidade 23 (menor número de movimentos possível para chegar da configuração inicial à configuração final).

A segunda configuração inicial não leva à configuração final proposta. No seu relatório explique o porque.

Apesar de propor apenas uma configuração inicial e final para teste, prepare o seu programa de forma a ser possível entrar com diversas configurações iniciais e finais. Portanto, escreva o seu código de forma que o utilizador possa escolher as configurações inicial e final. O seu código deve verificar se há solução para chegar do estado inicial ao estado final **antes** de iniciar a busca, e deve emitir uma mensagem de erro se não houver caminho entre a solução inicial e a final.

A configuração final de teste deve ser a seguinte:

1	2	3
8		4
7	6	5

Utilize o algoritmo 1 como base para implementar **todas** as buscas:

Algorithm 1 Algoritmo Geral de Busca

```
1: GeneralSearchAlgorithm(QueueingFunction)
2: queue = configInicial
3: while queue not empty do
4:   node = removeFrontNodeFrom(queue)
5:   if node is solution then
6:     return Path to solution
7:   end if
8:   descendantList = MakeDescendants(node)
9:   insert(descendantList,queue,QueueingFunction)
10: end while
11: return "solution not found"
```

Este algoritmo recebe como parâmetro uma função de enfileiramento que será utilizada para ordenar a lista de nós (configurações) abertos (ainda não explorados).

O que entregar?

1. trabalho escrito com o estudo das estratégias;

Organização do trabalho escrito e conteúdo (o template será disponibilizado na página da disciplina):

(a) Introdução

- O que é um problema de busca/procura?
- Quais são os métodos utilizados para resolver problemas de procura?
- Organização do trabalho (No capítulo 2 apresentamos os fundamentos...No Capítulo 3...)

(b) Estratégias de Procura

- i. Procura não guiada (blind - "cega")

- Profundidade (DFS - Depth-First Search: como funciona, quando se aplica, qual é a complexidade temporal e espacial?)
 - Largura (BFS - Breadth-First Search: como funciona, quando se aplica, qual é a complexidade temporal e espacial?)
 - Busca Iterativa Limitada em Profundidade (como funciona, quando se aplica, qual é a complexidade temporal e espacial?)
- ii. Procura guiada (que usa alguma função **heurística** para orientar a procura)
- Gulosa
 - Como funciona e quando se aplica?
 - Qual foi a heurística utilizada para o problema a ser resolvido e por que esta heurística foi escolhida?
 - Busca A*
 - Como funciona e quando se aplica?
 - Qual foi a heurística utilizada para o problema a ser resolvido e por que esta heurística foi escolhida?

(c) Descrição da Implementação

- Linguagem utilizada? Por que escolheu esta linguagem? Há alguma vantagem em utilizar esta linguagem para resolver este tipo de problema?
- Estruturas de dados utilizadas? Como foi que escolheu as estruturas de dados? São eficientes para manipular os dados do problema?
- Estrutura do código

(d) Resultados

Incluir tabela (ou curvas comparativas) com tempos de execução, utilização de memória e se encontrou a solução, para cada configuração, para cada estratégia, além da profundidade da solução encontrada. Se preferir utilizar uma tabela, esta deverá ter um sumário dos resultados organizados da seguinte forma:

Estratégia	Tempo (segundos)	Espaço	Encontrou a solução?
DFS
BFS
IDFS
Gulosa
A*

(e) Comentários Finais e Conclusões

Comentar sobre as estratégias fazendo uma comparação entre o seu desempenho e eficácia para encontrar as soluções. Concluir dizendo qual foi a melhor estratégia para este problema.

- (f) Referências Bibliográficas (precisam ser citadas no texto para saberem de onde o texto foi retirado/adaptado! Copiar é crime punido por lei, portanto evitem copiar textos de outros. O ideal é ler textos de vários autores, reescrever com suas próprias palavras e dar a sua própria interpretação, mas sempre citando as fontes de onde retiraram as ideias.)

Se utilizar figuras retiradas da web ou de livros ou de artigos etc, é necessário colocar uma referência.

Por favor, mantenham os erros ortográficos num nível mínimo.

2. Enviar através do Moodle um arquivo zip ou similar contendo o código fonte dos programas, e instruções de como compilar e executar cada problema, isto é, um pequeno manual de como correr os programas (pode ser um 'help' ou um 'readme'). Além disso, devem incluir uma pequena documentação explicando em que ambiente seu programa foi compilado (tipo e versão do SO e da linguagem). Seu programa deve correr na minha máquina (com fedora core 18 instalado). Não assumam que eu tenho uma IDE (Integrated Development Environment) de qualquer tipo. O programa deve correr na linha de comando.

O trabalho pode ser feito em grupo de **no máximo duas pessoas**. Todos os trabalhos deverão ser apresentados em data a combinar. **Todos os componentes do grupo deverão estar presentes durante a demonstração. Um dos componentes do grupo será aleatoriamente escolhido para responder às perguntas formuladas.** Quem não estiver presente vai ter nota zero! Cada componente do grupo deverá comentar sobre sua contribuição no trabalho.

Se o trabalho for dividido entre os componentes do grupo, assegurem-se de que cada um sabe exatamente o que o outro está a fazer para evitar constrangimentos e penalizações durante a sessão de apresentação.