

Primeiro Trabalho de Sistemas Inteligentes

Entrega: 07/03/2014

21 de Fevereiro de 2014

Este trabalho vale nota e será utilizado para calcular a média 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 formando uma imagem completa, no caso da matriz onde as células são parte de uma imagem. Os movimentos/operações 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:

- complexidade temporal (tempo para chegar a uma solução),
- complexidade espacial (número de nós armazenados),
- completude (o algoritmo consegue encontrar uma solução?) e
- otimalidade (o algoritmo consegue encontrar a solução no menor número de passos em tempo hábil e utilizando pouca memória? Qual a profundidade da solução?)

Utilize as seguintes configurações iniciais:

3	2	4
5	1	8
	6	7

5	7	2
6		3
8	4	1

A configuração final deve ser a seguinte:

1	2	3
8		4
7	6	5

O seu programa deve verificar se há solução para chegar do estado inicial ao estado final **antes** de iniciar a busca.

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 (o template será disponibilizado na página da disciplina):

(a) Introdução

- O que são algoritmos de busca não informados e para que servem.
- O que são algoritmos de busca informados e para que servem.
- Organizaçao do trabalho (No captulo 2 apresentamos os fundamentos...No Captulo 3....)

(b) Estratégias

- Busca em Profundidade (O que é e quando se aplica)
- Busca em Largura (O que é e quando se aplica)
- Busca Iterativa Limitada em Profundidade (O que é e quando se aplica)
- Busca Gulosa
 - O que é e quando se aplica
 - Qual foi a heurística utilizada para cada problema resolvido e por que esta heurística foi escolhida?
- Busca A*
 - O que é e quando se aplica

- Qual foi a heurística utilizada para cada problema resolvido e por que esta heurística foi escolhida?
 - (c) Resultados
 - Incluir tabela com tempos de execução, utilização de memória e se encontrou a solução, para cada configuração, para cada estratégia, além da profundidade da solução encontrada.
 - (d) 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.
 - (e) Referências Bibliográficas (precisam ser citadas no texto!)
2. Enviar através do Moodle um arquivo zip ou similar contendo o código fonte dos programas, instruções de como compilar e executar cada problema, isto , um pequeno manual de como correr os programas (pode ser um 'help' ou 'readme'). Alm 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 assuma 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 as perguntas.** 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.