Departamento de Informática - UEM
Disciplina: Modelagem e Otimização Algorítmica
2ª Avaliação de 2017
Nome: _____

Descrição.
Implementação do algoritmo A* para a resolução do Jogo do Tabuleiro de 15 peças. A implementação deverá permitir entrar com qualquer estado inicial e o objetivo é encontrar a menor sequência de movimentos que leve ao estado final do jogo. Estado final desejado é apresentado a seguir:

Estado final desejado:

1	2	3	4
12	13	14	5
11		15	6
10	9	8	7

Heurísticas a serem implementadas e analisadas:

1. $h'_1(n)$ = número de peças foras de seu lugar na configuração final.
2. $h'_2(n)$ = número de peças fora de ordem na sequência numérica das 15 peças, seguindo a ordem das posições no tabuleiro.
3. $h'_3(n)$ = para cada peça fora de seu lugar somar a distância Manhattan (quantidade de deslocamentos) para colocar em seu devido lugar. Neste caso considera-se que o caminho esteja livre para fazer o menor número de movimentos.
4. $h'_4(n) = p_1 * h'1(n) + p_2 * h'2(n) + p_3 * h'3(n)$, sendo $p_1 + p_2 + p_3$ são pesos (número real) tais que $p_1 + p_2 + p_3 = 1$. A escolha desses pesos deverão ser escolhidos conforme os resultado dos experimentos.
5. $h'_5(n) = \max(h'_1(n), h'_2(n), h'_3(n))$.

A avaliação se dará em duas partes, sendo que cada parte corresponde a 50% da nota.

Parte 1 (individual**)**: Submissão e correção dos trabalhos pelo sistema Run.Code (https://run.codes/). O sistema só aceita código nas seguintes linguagens de programação: **Java 8, C/C++, Python 3, Pascal, e Fortran90**. Detalhes de como utilizar o sistema pode ser obtidos em support@run.codes. Cadastre-se no sistema com o código: K46N. Esta parte da avaliação deve ser realizada individualmente. Fique atentos ao sistema pois esta parte será dividida em algumas fases com datas de submissão com diferentes datas de submissão. O sistema Run.Codes realiza tanto testes de execução de seu código como, também, avaliação de similaridade do seu código fonte com uma base de outros códigos.

Parte 2 (equipe com 2 membros): Relatório impresso, incluindo a análise do comportamento das 5 propostas de heurísticas de todos os casos disponibilizados pelo professor no Moodle. Nesta análise deve levar em consideração o consumo de memória e o tempo de processamento. Considerar todos os casos disponibilizados no *Moodle* especificamente para este relatório.

Prazo para entrega: até 19/11/2017.

Exemplo de como calcular a heurística h'$_2$.

2	1	5	13
0	3	4	14
8	10	9	15
7	6	12	11

Primeiro passo considerar a sequências de "posições" no tabuleiro conforme a sequência "numérica" esperada para o estado final, ou seja, iniciando na posição [1,1] (linha 1 e coluna 1) e terminando na posição [3,2] (linha 3, coluna 2).

A partir da tabela acima, teremos a seguinte sequência:

2	1	5	13	14	15	11	12	6	7	8	0	3	4	9	10

Agora é verificada a sequência numérica. As posições em amarelo são posições contabilizadas para esta heurística. Conforme se observa, as ocorrências contabilizadas são estas:

1. Peça 1 após a peça 5, enquanto esperava a peça 6;
2. Peça 5 após a peça 1;
3. Peça 13 após a peça 5;
4. Peça 11 após a peça 15;
5. Peça 6 após a peça 12;
6. Espaço vazio após a peça 8;
7. Peça 9 após a peça 4.

Note que após o espaço vazio não é feita a contabilização.

Portanto, para este caso h'2=7.