

Trabalho Prático 1 - Ordenação usando Busca em Espaço de Estados

Data de entrega: 15/05

Você acabou de iniciar um novo estágio em uma grande empresa. Sua primeira tarefa é implementar um programa para ordenar um vetor com n números inteiros variando de 1 a n , sem repetição. Parece simples, mas infelizmente você não assistiu as aulas sobre ordenação em Estrutura de Dados, e agora está em apuros... Por sorte, você acabou de estudar o conteúdo de Busca em Espaço de Estados em Inteligência Artificial, e teve a brilhante ideia de resolver esse problema usando busca!

Basicamente, você vai começar com um vetor aleatório e ir fazendo trocas de elementos, um de cada vez, até finalizar com o vetor ordenado. A máquina que você trabalha tem algumas características bem específicas: a troca entre elementos vizinhos tem custo 2, enquanto trocas entre elementos que não são vizinhos tem custo 4. Você quer encontrar a solução de menor custo e para isso vai testar os seguintes algoritmos:

- *Breadth-first search*
- *Iterative deepening search*
- *Uniform-cost search*
- *A* search*
- *Greedy best-first search*

Com relação aos algoritmos A* e Greedy best-first, você deve propor e implementar uma heurística adequada.

Implementação e Formato de Entrada e Saída:

O trabalho pode ser implementado em uma das seguintes linguagens: C, C++ ou Python.

Para facilitar a correção, o seu programa deverá se chamar TP1 e ler os parâmetros a partir da linha de comando. O primeiro parâmetro deverá ser o algoritmo a ser utilizado (B, I, U, A, G), seguido do tamanho do vetor n e os seus elementos. E, opcionalmente, um último parâmetro (PRINT) indicando se os vetores intermediários da solução encontrada devem ser impressos.

Por exemplo, para ordenar o vetor 5 3 2 4 1 usando o A*, a sua entrada deverá ser:

```
% TP1 A 5 5 3 2 4 1
```

A saída deverá ser o custo da solução encontrada e o número de estados expandidos pelo algoritmo. Se a opção PRINT for utilizada, você deverá imprimir também todos os estados intermediários até a solução. Por exemplo, para a entrada acima, deve ser impresso algo como

```
% TP1 A 5 5 3 2 4 1 PRINT
6 2
5 3 2 4 1
5 2 3 4 1
1 2 3 4 5
```

O que deve ser entregue:

Faça um .zip ou similar contendo:

- 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:

- Apresentação das estruturas usadas e da modelagem dos componentes da busca (estado, função sucessora, etc);
- Breve descrição das principais diferenças entre os algoritmos;
- Especificação da heurística utilizada. Ela é admissível? Por quê?;
- Exemplos de soluções encontradas pelos algoritmos;
- Análise quantitativa comparando os algoritmos com relação ao número de estados expandidos e tempo de execução para diferentes tamanhos de entrada (por exemplo, vetores de tamanho 5, 10, 15 e 20). **Apresente tabelas e/ou gráficos comparativos**;
- Discussão dos resultados obtidos.

Bom trabalho!