

Primeiro Exercício Programa (EP1)

Problema: Travessia da Ponte

Data de entrega: 1/10/2014

Composição dos grupos: de 3 a 4 alunos.

A banda de rock U2 tem um concerto que será realizado no lado Leste da cidade e que começa em MAX minutos. Para assistir ao concerto, um grupo de pessoas precisam atravessar caminhando uma ponte que liga o lado Oeste ao Leste da cidade. Está escuro, e só há uma lanterna. Além disto, a ponte suporta no máximo duas pessoas por vez, ou seja, as travessias podem ser feitas por 1 ou 2 pessoas de cada vez. Quem for atravessar a ponte, deve passar com a lanterna na mão. A lanterna deve ser levada de um lado para o outro, não podendo ser descartada. Cada pessoa do grupo é capaz de atravessar a ponte em um determinado tempo, porém, em uma travessia, o par deve caminhar no tempo do menos veloz. O problema consiste em ter o grupo de pessoas no outro lado da ponte no menor tempo possível e que necessariamente seja antes do início do concerto, isto é, em MAX minutos.

Por exemplo, suponha que o tempo para iniciar o concerto é $MAX=17$ e que temos um grupo de 4 pessoas, Bono, Edge, Adam e Larry, sendo que:

- Bono atravessa a ponte em 1 minuto;
- Edge atravessa a ponte em 2 minutos;
- Adam atravessa a ponte em 5 minutos; e
- Larry atravessa a ponte em 10 minutos.

É possível que eles cheguem a tempo para assistirem o concerto do U2? Se sim, qual é a sequência de travessias que permite que o grupo chegue a tempo ao concerto? Qual é a sequência que possui o menor número de travessias, ainda que o grupo chegue atrasado para o concerto?

1. Especificação do Exercício Programa

O trabalho consiste em implementar soluções para esse problema de travessia usando os algoritmos de busca vistos em sala de aula:

- 1) busca em largura,
- 2) busca em profundidade,
- 3) busca em profundidade iterativa,
- 4) busca de custo uniforme,

- 5) busca gulosa, e
- 6) busca A*

Considere o tempo de travessia como o custo das ações e $g(n)$ como o tempo total do caminho da solução. Para os algoritmos 4 e 6, a solução não deve exceder o tempo para o início do concerto, isto é, MAX. Para os demais algoritmos de busca, ao invés de devolver a solução que não excede o tempo para o início do concerto, isto é, MAX, seu algoritmo deve devolver a solução com menor número de travessias (profundidade) (algoritmos 1 e 3) ou a primeira solução encontrada (algoritmo 2 e algoritmo 5). Para os algoritmos de busca informada, você deve usar a heurística h_1 :

h_1 = o menor tempo de travessia das pessoas que estão do lado Oeste da ponte.

Além disso, você deve projetar mais duas heurísticas diferentes, h_2 e h_3 , juntamente com uma explicação informal sobre a admissibilidade das 3 heurísticas .

O seu programa deve aceitar 3 instâncias desse problema de diferentes tamanhos e devolver a mensagem “*solução não encontrada*”, caso o algoritmo não encontre uma solução. Caso contrário, o algoritmo deve devolver a sequência de passos, o custo total da solução encontrada, a profundidade da solução, o tempo em milissegundos gasto pelo algoritmo e o número total de estados visitados (tamanho do conjunto explorado).

2. Problemas a serem resolvidos

Instância 1: (problema descrito na introdução) Um grupo de 4 pessoas, com MAX=17 minutos, e tempos de travessia: (1, 2, 5, 10).

Instância 2: um grupo de 10 pessoas, com MAX=104 minutos, e tempos de travessia: (1, 2, 5, 10, 3, 4, 14, 18, 20, 50).

Instância 3: um grupo de 15 pessoas, com MAX=182 minutos, e tempos de travessia: (1, 2, 5, 10, 12, 17, 24, 21, 20, 20, 11, 33, 15, 19, 55).

3. Entradas

A entrada do seu programa é um arquivo de texto que está dividido em duas partes, separadas por uma linha em branco. A primeira parte tem três números: b , n e MAX que indicam o tipo de busca escolhida, a quantidade total de pessoas e o tempo máximo aceitável da solução, respectivamente. No caso de busca heurística essa primeira parte também deverá conter o nome da função heurística usada.

A segunda parte do arquivo de entrada contém n linhas com o tempo (em minutos) necessário para cada uma das n pessoas atravessar a ponte (note que uma pessoa é identificada pelo tempo que ela gasta na travessia).

Por exemplo, a especificação de entrada para a Instância 1 (problema descrito na introdução) usando a busca em largura, é:

=====instância1-1.txt

1 4 17

1
2
5
10

4. Formato de Saída

A solução encontrada deve ser devolvida como uma sequência de travessias, cada uma indicada pelos tempos de travessia das pessoas (1 ou 2 pessoas), seguidos de ">>", se a travessia foi feita do Oeste para Leste ou "<<", se a travessia foi feita do Leste para Oeste. Cada travessia deve estar em uma linha diferente. Em seguida, após uma linha em branco, deve ser impresso em uma mesma linha (com espaços em branco separando):

- o custo total (tempo total gasto nas travessias),
- a profundidade da solução,
- o tempo em mili-segundos gasto pelo algoritmo e
- o número total de estados visitados (tamanho do conjunto explorado).

Por exemplo uma solução para a Instância 1 poderia ser:

1 2 >>
1 <<
5 10 >>
2 <<
1 2 >>

17 5 230 30

A solução acima tem 5 travessias, consumindo um total de 17 minutos, encontrada na profundidade 5, consumiu 230 mili-segundos de execução do algoritmo e a busca visitou um total de 30 estados.

5. Data de entrega: 1/10/2014

A entrega do programa no Tidia (incluindo arquivos fonte) e do relatório de no máximo 10 páginas, deve ser feita até o final do dia **1/10/2014**.. Não serão aceitos envios por email.

O relatório deve incluir:

- um diagrama de classes,
- a descrição das heurísticas implementadas especificando se são admissíveis ou não,
- resultados dos experimentos realizados com as 3 instâncias do problema e todos os algoritmos implementados. O relatório deve incluir uma descrição dos experimentos, tabelas, gráficos e discussão dos resultados obtidos.