

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Algoritmo e Estruturas de Dados II (DCC004)
Trabalho Prático II

O caminho mais rápido

Entrega: x/Jun/2016 via Moodle

Este trabalho prático tem por objetivo familiarizar o aluno com algoritmos de caminhamento em árvores, além de desenvolver a capacidade em implementar um tipo abstrato de dados eficaz, com todas as funções necessárias para resolver o problema proposto.

1 Definição do Problema

Você é fundador de uma startup que está começando a crescer rapidamente e já tem um número razoável de funcionários. Nos últimos meses porém, você verificou no banco de horas de seus empregados que muitos chegam atrasados. Incomodado com a situação, você faz uma pesquisa interna e descobre que a grande maioria se desloca de carro ao escritório e o trânsito é a principal explicação para o problema dos atrasos constantes. Numa maneira de motivar seus funcionários e ao mesmo tempo diminuir o problema, você consegue um mapa das ruas da cidade com o tempo que um carro costuma gastar para atravessar cada uma das ruas durante o período de engarrafamento. O seu objetivo é de criar um algoritmo que dado um mapa com as ruas e o tempo necessário para atravessar cada uma, calcule o tempo mínimo para se ir de um determinado local de origem ao local de destino para distribuir a todos os seus funcionários.

2 Entrada do Algoritmo

A entrada é composta de vários casos de teste distintos e é representada da seguinte maneira:

- Inicia-se com dois inteiros **n** e **m**, representando respectivamente o número de ruas da cidade e o número de ruas a serem consideradas na solução.
- Logo em seguida, temos **m** linhas com mais três inteiros **i**, **j** e **t**. O par **i**, **j** representa que existe um caminho de mão **única** entre as respectivas ruas ¹ onde **t** é o tempo médio que um veículo leva para percorrer este caminho.
- A última linha contém mais um par de inteiros **s** e **d** representando respectivamente os pontos de origem e destino da trajetória.

¹No caso de existir um caminho de mão dupla entre as ruas **i** e **j**, ele será representado por dois pares: (**i**,**j**) e (**j**,**i**).

- Seu algoritmo deve parar no momento em que **m** e **n** forem iguais a 0.

Exemplo do arquivo de entrada tp2.in:

```
3 3
1 2 2
1 3 7
2 3 3
1 3
3 2
1 2 2
2 1 3
1 3
4 6
1 2 5
1 3 3
2 4 3
3 2 2
4 1 4
4 3 9
2 3
0 0
```

3 Saída do Algoritmo

Para cada teste, o seu algoritmo deve imprimir uma linha com um inteiro com o tempo mínimo necessário para se deslocar da origem **s** ao destino **d**. No caso de não ser possível alcançar **d** a partir de **s**, a linha deverá imprimir o inteiro **-1**.

Exemplo do arquivo de saída tp2.out:

```
5
-1
10
```

4 Implementação para Pontos Extras (x pts)

Depois de calcular os caminhos mais rápidos para seus funcionários, você, no seu tempo livre decide verificar quais desses caminhos são os que ainda gastam mais tempo. Implemente um algoritmo de ordenação de sua escolha (sem o uso de bibliotecas externas) que, após o cálculo dos casos de testes no arquivo de entrada, ordene os caminhos de maneira **decrecente** em relação ao tempo calculado. **OBS.:** *Os caminhos calculados como -1 não devem ser considerados.*

O arquivo de saída conterá uma linha com o valor do caminho mínimo encontrado, seguido de uma linha com os dois inteiros **s** e **d**, representando os pontos de partida e destino respectivamente para cada caso de teste da entrada.

Exemplo do arquivo de saída xtp2.out:

Considere para o mesmo exemplo de entrada da seção 2 (tp2.in)

```
10
2 3
5
1 3
```

5 Restrições

- A implementação deve ser feita obrigatoriamente em C sem a utilização de quaisquer bibliotecas externas.
- Sua estrutura de dados deverá ser implementada dinamicamente.
- A implementação deverá compilar sem *warnings* em ambiente Linux (Ubuntu 14.04.LTS), utilizando a seguinte linha de comando:

```
gcc -Wall *.c -o EP2
```

- O programa deverá rodar recebendo o arquivo de entrada tp2.in como argumento passado na linha de comando:

```
./TP2 tp2.in
```

- Seu programa deve gerar um arquivo de saída com o nome tp2.out. No caso de fazer a implementação elegível para pontos extras, serão dois arquivos gerados: tp2.out e xtp2.out
- A saída deve obedecer rigorosamente ao padrão especificado pois será verificada de maneira automática e representará a maior parte da nota.

6 Entregáveis

Crie um diretório (nomeie o diretório com seu número de matrícula) contendo seu código fonte (todos os arquivos .c e .h) e comprima-o com o seguinte comando:

```
tar zcvf <diretorio>.tar.gz <diretorio>
```

onde <diretorio> indica o nome do diretório criado. Submeta o arquivo .tar.gz via Moodle.

7 Critérios de Correção

A pontuação do trabalho se divide nos seguintes critérios:

- (10%) Adequação às normas de envio.
- (20%) Funcionamento correto do programa no *valgrind*, ausência de erros de execução (e.g.: *segmentation fault*), validação da entrada (e.g.: checagem se o número de argumentos passados está correto).
- (70%) Saída correta do programa.