

INSTITUTO SUPERIOR TÉCNICO  
**Análise e Síntese de Algoritmos**  
**2014/2015**

**2º Projeto**

Data Limite de Entrega: 24 de Abril de 2015

## **O Problema**

A empresa Coelho e Caracol Lda. faz transporte de mercadorias. O Sr. Coelho passa o dia na camioneta a andar de local para local, enquanto o Sr. Caracol fica no escritório a fazer o planeamento das rotas. No entanto, devido à expansão da empresa, a tarefa do Sr. Caracol está mais difícil. A quantidade de rotas a planear é cada vez maior. Neste projeto iremos desenvolver software para resolver este problema automaticamente.

Uma rota consiste numa sequência de localidades, sendo que entre cada par de localidades pode ser transportado um tipo de mercadoria. O transporte de mercadorias gera receita, enquanto que a deslocação tem um custo, combustível, portagens, etc.

O Sr. Caracol simplifica esta complexidade associando a cada par de localidades um valor de perda, que resulta de subtrair a receita ao custo. Portanto, se a receita for maior que o custo, o valor será negativo. Se o valor for positivo significa que o custo supera a receita. Naturalmente, o Sr. Caracol prefere ter valores negativos entre duas localidades. Nesse sentido, alinha a estratégia da empresa por forma a que esses valores sejam o menor possível, sendo que entre um determinado par de localidades é considerada no máximo um valor de perda.

Feita esta seleção inicial, o objetivo é estabelecer para cada localidade qual é o percurso que, com origem na sede da empresa, minimiza o valor de perda total. Ao fazer esta análise o Sr. Caracol tem tristezas e alegrias. De vez em quando constata que não é possível chegar a uma determinada localidade. Por outro lado, às vezes constata, com muita alegria, que há uma sequência de localidades que, se iterada um número suficientemente grande de vezes, produz um ganho tão grande quanto se queira.

## Input

O ficheiro de entrada deverá conter a informação sobre as perdas entre localidades. O input é definido da seguinte forma:

- Uma linha com o número de localidades  $N$  ( $N \geq 2$ ) e o número de custos conhecidos  $C$ , separados por um espaço em branco.
- Uma linha com o número entre 1 e  $N$  que identifica a sede da empresa.
- Uma lista de  $C$  linhas, em que cada linha contém três inteiros  $u$ ,  $v$  e  $w$  (separados por um espaço em branco). Cada linha indica que a deslocação de  $u$  para  $v$  tem perda  $w$ .

Assume-se que a identificação das localidades é um inteiro entre 1 e  $N$ .

## Output

Para cada localidade queremos determinar a perda mínima até esse ponto, partindo da sede. O programa deverá escrever no output a seguinte informação:

- Uma sequência de  $N$  linhas em que a linha  $i$  ( $1 \leq i \leq N$ ) corresponde à perda mínima do ponto  $i$ . Caso seja impossível definir essa perda, a linha deve conter apenas o carácter 'I'. Caso não existam deslocamentos suficientes para determinar essa perda deve conter apenas o carácter 'U'.

## Exemplos

### input 1

```
9 8
7
2 3 -100
3 4 -100
1 2 -100
4 5 -100
5 1 -300
7 8 100
5 6 400
7 1 100
```

### **output 1**

I  
I  
I  
I  
I  
I  
I  
0  
100  
U

### **input 2**

6 8  
1  
1 2 -100  
2 3 -100  
1 3 -198  
3 4 -100  
1 4 -280  
4 5 -100  
5 1 450  
6 1 100

### **output 2**

0  
-100  
-200  
-300  
-400  
U

### **input 3**

```
6 8
1
1 2 100
2 3 100
1 3 198
3 4 100
1 4 280
4 5 100
5 1 -350
6 1 -100
```

### **output 3**

```
0
100
198
280
380
U
```

## **Implementação**

A implementação do projeto deverá ser feita preferencialmente usando as linguagens de programação C ou C++. Submissões em linguagem Java também são aceitáveis, mas devem ter mais cuidado com alguns aspetos de implementação.

O tempo necessário para implementar este projeto é inferior a 20 horas.

## **Submissão do Projeto**

A submissão do projeto deverá incluir um relatório resumido e um ficheiro com o código fonte da solução. Informação sobre as linguagens de programação possíveis está disponível no website do sistema Mooshak. A linguagem de programação é identificada pela extensão do ficheiro. Por exemplo, um projeto escrito em C deverá ter a extensão .c. Após a compilação, o programa resultante deverá ler do 'standard input' e escrever para o 'standard output'. Informação sobre as opções e restrições de compilação podem ser obtidas através do botão 'help' do sistema

Mooshak. O comando de compilação não deverá produzir output, caso contrário será considerado um erro de compilação. O relatório deverá ser entregue no formato PDF com não mais de 4 páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma introdução breve, a descrição da solução, a análise teórica e a avaliação experimental dos resultados. O relatório deverá incluir qualquer referência que tenha sido utilizada na realização do projeto. Relatórios que não sejam entregues em formato PDF terão nota 0. O código fonte deve ser submetido através do sistema Mooshak e o relatório (em formato PDF) deverá ser submetido através do Fénix. O código fonte será avaliado automaticamente pelo sistema Mooshak. Observe que apenas a última submissão será considerada para efeitos de avaliação. Todas as submissões anteriores serão ignoradas; tal inclui o código fonte e o relatório.

Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak e para o Fénix. Note que também é possível submeter várias vezes no Fénix e que não serão aceites relatórios fora de prazo e não haverá extensão de prazo.

O sistema Mooshak indica o tempo disponível para o projeto ser submetido. Os projetos têm que ser submetidos para o sistema Mooshak; não existe outra forma de submissão do projeto. Os relatórios têm que ser submetidos no sistema Fénix; não existe outra forma de submissão dos relatórios.

## Avaliação

O projeto deverá ser realizado em grupos de um ou dois alunos e será avaliado em duas fases. Na primeira fase, durante a submissão, cada implementação será executada num conjunto de testes, os quais representam 80% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 20% da nota final.

### Avaliação Automática

A primeira fase do projeto é avaliada automaticamente com um conjunto de testes, os quais são executados num computador com o sistema operativo **GNU/Linux**. É essencial que o código fonte compile sem erros e respeite os standards de entrada e saída indicados anteriormente. Os projetos que não respeitem os formatos especificados serão penalizados e poderão ter nota 0, caso falhem todos os testes. Um conjunto reduzido de testes utilizados pelo sistema Mooshak será publicado na página da disciplina. A maior parte dos testes **não** serão divulgados antes da submissão. No entanto, todos os testes serão disponibilizados após o deadline para submissão do projeto. Além de verificar a correção do output produzido, o ambiente de avaliação restringe a memória e o tempo de execução disponíveis. A maior parte dos testes executa o comando `diff` da forma seguinte:

```
diff output result
```



O ficheiro `result` contém o output gerado pelo executável a partir do ficheiro `input`. O ficheiro `output` contém o output esperado. Um programa passa num teste e recebe o valor correspondente, quando o comando `diff` não reporta quaisquer diferenças (i.e., não produz qualquer output). Existem 16 testes. Assim, o sistema reporta um valor entre 0 e 16.

### **Deteção de Cópias**

A avaliação dos projetos inclui um procedimento para deteção de cópias. A submissão de um projeto implica um compromisso de que o trabalho foi realizado exclusivamente pelos alunos. A violação deste compromisso ou a tentativa de submeter código que não foi desenvolvido pelo grupo implica a reprovação na unidade curricular, para todos os alunos envolvidos (incluindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, direta ou indireta, será comunicada ao Conselho Pedagógico do IST, ao coordenador de curso, e será penalizada de acordo com as regras aprovadas pela Universidade e publicadas em “Diário da República”.

O ficheiro `result` contém o output gerado pelo executável a partir do ficheiro `input`. O ficheiro `output` contém o output esperado. Um programa passa num teste e recebe o valor correspondente, quando o comando `diff` não reporta quaisquer diferenças (i.e., não produz qualquer output). Existem 16 testes. Assim, o sistema reporta um valor entre 0 e 16.

### **Deteção de Cópias**

A avaliação dos projetos inclui um procedimento para deteção de cópias. A submissão de um projeto implica um compromisso de que o trabalho foi realizado exclusivamente pelos alunos. A violação deste compromisso ou a tentativa de submeter código que não foi desenvolvido pelo grupo implica a reprovação na unidade curricular, para todos os alunos envolvidos (incluindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, direta ou indireta, será comunicada ao Conselho Pedagógico do IST, ao coordenador de curso, e será penalizada de acordo com as regras aprovadas pela Universidade e publicadas em “Diário da República”.