INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos 2015/2016

2º Projecto

Data Limite de Entrega: 22 de Abril de 2016

O Problema

A empresa Coelho e Caracol Lda. faz transporte de mercadorias. Esta empresa teve um enorme sucesso no ano passado devido ao empenho dos alunos de Análise e Sintese de Algoritmos do IST. O Sr. Caracol fica no escritório a fazer o planeamento das rotas, enquanto o Sr. Coelho trata da coordenação das camionetas entre as várias filiais existentes. Para celebrar o sucesso da empresa o Sr. Caracol decidiu organizar um encontro de empresa, que deverá juntar os vários empregados das diversas filiais. O problema consiste em encontrar um ponto de encontro adequado. Neste projeto iremos desenvolver software para resolver este problema automaticamente.

Para minimizar os custos das rotas, das diversas filiais, até ao ponto de encontro os empregados podem fazer entregas ao longo do percurso. 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. Visto que todas as filiais trabalham para a mesma empresa estes valores são uniformes para todas as filiais, i.e., não variam.

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.

Cada filial establece para cada localidade qual é o percurso que, com origem nessa filial, minimiza o valor de perda total. Visto que existem várias filiais o Sr. Caracol pretende descobrir

qual é a melhor localidade para organizar o encontro, ou seja qual é a localidade em que a perda total da empresa, considerando todas as filiais, é minimizada.

Assuma que este problema tem as seguintes propriedades: o número de filiais é significativamente menor que o número de localidades; o número de pares de localidades entre os quais há um valor de perda é muito menor que o número de todos os pares possíveis; não existe um percurso entre localidades que forme um ciclo com valor de perda total negativo.

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 ($N \geq 2$), o número de filiais F ($F \geq 2$) e o número de ligações C. Estes valores estão separados por um espaço em branco.
- Uma linha com F números entre 1 e N que identifica as filiais.
- 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

Queremos determinar a localidade que deverá ser utilizada como ponto de encontro, e a respetiva perda total da empresa. Para cada filial queremos também saber qual é a respetiva perda e rota.

- Um linha contendo o número que identifica o ponto de encontro e a respetiva perda total, separados por um espaço em branco.
- Uma segunda linha com os valores de perda mínima de cada uma das filiais i ($1 \leq i \leq F$) até ao ponto de encontro, separados por espaços. As filiais devem ser consideradas pela mesma ordem do input. Existe um espaço em branco antes do caracter de fim de linha.
- Caso não seja possível encontrar um ponto de encontro, o output consiste numa única linha com a letra `N`. Esta situação apenas pode ocorrer quando alguma filial está "isolada" das restantes.

Exemplos

input 1

```
5 2 8
1 2
4 5 -5
2 4 3
1 4 2
3 4 -4
1 5 -1
2 3 5
1 3 4
2 5 -2
```

output 1

```
5 -9
-5 -4
```

input 2

```
6 3 11
1 5 2
1 2 -5
6 4 3
1 3 -2
5 3 4
2 4 2
4 2 2
3 4 -3
2 3 3
3 5 3
4 6 2
5 6 -4
```

output 2

```
4 -6
-5 -1 0
```

input 3

```
6 2 11
1 5
1 2 -5
6 4 3
1 3 -2
5 3 4
2 4 2
4 2 2
3 4 -3
2 3 3
3 5 3
4 6 2
5 6 -4
```

output 3

```
6 -7
-3 -4
```

input 4

```
5 3 6
1 2 5
1 5 -5
2 4 3
1 4 2
3 4 -4
2 3 5
1 3 4
```

output 4

```
N
```

Implementação

A implementação do projecto 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 aspectos de implementação.

O tempo necessário para implementar este projecto é inferior a 25 horas.

Submissão do Projecto

A submissão do projecto 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 projecto 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 projecto. 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 possivel 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 projecto ser submetido. Os projectos têm que ser submetidos para o sistema Mooshak; não existe outra forma de submissão do projecto. Os relatórios têm que ser submetidos no sistema Fénix; não existe outra forma de submissão dos relatórios.

Avaliação

O projecto 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 projecto 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 projectos 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 serão públicos. A maior parte dos testes **não** serão divulgados antes da submissão. No entanto, todos os testes serão disponibilizados após o deadline para submissão do projecto. Além de verificar a correcção do output produzido, o ambiente de avaliação restringe a mémoria 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.

Detecção de Cópias

A avaliação dos projectos inclui um procedimento para detecção de cópias. A submissão de um projecto 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 (includindo os alunos que disponibilizaram o código). Qualquer tentativa de fraude, directa or indirecta, 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".