

# **Análise e Síntese de Algoritmos 2004/05**

## **Instituto Superior Técnico**

### **Enunciado do 2º Projecto**

**Data de Entrega: 13 de Maio de 2005**

#### **1 Introdução**

O 2º projecto de ASA tem como objectivo a realização de um programa em C++ que calcula actualizações ao valor do fluxo máximo de uma rede de fluxo.

#### **2 Especificação do Problema**

Seja  $G = (V, E)$  uma rede de fluxo, com vértice fonte  $s$  e vértice destino  $t$ , em que cada arco  $(u, v)$  tem capacidade com valor inteiro  $c(u, v)$ . O fluxo  $f$ , correspondente ao fluxo máximo entre  $s$  e  $t$ , encontra-se calculado. Nestas condições, admita que as capacidades de um conjunto de arcos  $A = \{(x_1, y_1), \dots, (x_k, y_k)\}$  são alteradas. Para cada arco  $(x_i, y_i)$ , com  $1 \leq i \leq k$ , a capacidade é alterada do valor inicial  $c(x_i, y_i)$  para o novo valor  $c'(x_i, y_i)$ , tal que  $p_i = c(x_i, y_i) - c'(x_i, y_i)$  é um valor inteiro. Nestas condições pretende-se calcular o valor do fluxo máximo para a rede de fluxo com os novos valores de capacidade.

#### **3 Dados de Entrada**

O ficheiro de entrada deverá conter a informação seguinte:

- Uma linha com o número de vértices do grafo,  $N$ , e com o número de arcos do grafo,  $M$ . Observe que os vértices são numerados consecutivamente e que, por convenção, os vértices 1 e 2 representam respectivamente os vértices fonte e destino.
- Uma lista de  $M$  linhas, em que cada linha contém três inteiros (separados por um espaço em branco) os quais representam os vértices de cada arco e a respectiva capacidade. Os valores da capacidade são inteiros positivos. A ordem das linhas é qualquer.
- Uma linha com o carácter '#'.

- Uma lista de  $M$  linhas, em que cada linha contém três inteiros (separados por um espaço em branco) os quais representam os vértices de cada arco e o respectivo fluxo para o fluxo máximo  $f$ . A ordem das linhas é qualquer.
- Uma linha com o caracter '#'.
- Uma linha com um número  $K$ , indicando o número de arcos cujo valor de capacidade deverá ser alterado.
- Uma lista de  $K$  linhas, em que cada linha contém três inteiros (separados por um espaço em branco) os quais representam os vértices de cada arco e o novo valor de capacidade para o arco. O novo valor de cada capacidade é um inteiro não negativo. A ordem das linhas é qualquer.

## 4 Dados de Saída

O programa deverá escrever o novo valor do fluxo máximo.

## 5 Exemplos

### Dados de Entrada

```
6 7
1 3 12
1 4 10
3 5 11
4 6 11
5 6 1
5 2 9
6 2 11
#
1 3 10
1 4 10
3 5 10
4 6 10
5 2 9
5 6 1
6 2 11
#
1
5 6 2
```

### Dados de Saída

```
20
```

**Dados de Entrada**

```
6 7
1 3 12
1 4 10
3 5 11
4 6 11
5 6 1
5 2 9
6 2 11
#
1 3 10
1 4 10
3 5 10
4 6 10
5 2 9
5 6 1
6 2 11
#
2
5 6 2
1 4 9
```

**Dados de Saída**

```
20
```

## 6 Entrega do Projecto

A entrega do projecto deverá respeitar o procedimento indicado na página da disciplina. O processo de submissão apenas aceita projectos realizados em C ou C++. No entanto, recomenda-se a realização do programa na linguagem de programação C++.

## 7 Entrega do Projecto

A entrega do projecto deverá respeitar o procedimento seguinte:

- No site da disciplina deverá aceder à página para entrega de projectos.
- Deverá efectuar o *upload* de um ficheiro com proj2.cc.
- O compilador utilizado é o compilador g++ (versão 3.3.3) da GNU, sendo utilizada a opção '-ansi', por forma a requerer que o programa respeite a norma ANSI C.

- Como resultado do processo de *upload* será informado se a resolução entregue apresenta a resposta esperada num conjunto de casos de teste.

## 8 Avaliação do Projecto

### 8.1 Componentes da Avaliação

Na avaliação do projecto serão consideradas as componentes seguintes:

1. A primeira componente avalia o correcto funcionamento do programa realizado, e tem uma contribuição máxima de 16 valores para a nota final.
2. A segunda componente avalia o desempenho do programa realizado, e tem uma contribuição máxima de 4 valores para a nota final.
3. A terceira componente da avaliação corresponde à certificação com o programa *valgrind*. Projectos que reportem erros como resultado da certificação com o *valgrind* têm uma penalização de 3 valores.

### 8.2 Atribuição da Nota Relativa a Cada Componente

A nota de cada componente da avaliação do projecto é obtida automaticamente através da execução de um conjunto de testes executados num computador com o sistema operativo **GNU/Linux**. Os testes considerados para efeitos de avaliação incluem os disponibilizados na página da disciplina, além de um conjunto de testes adicionais.

### 8.3 Detecção de Cópias

A avaliação dos projectos inclui a utilização de um sistema para detecção de situações de cópia entre projectos. O sistema utilizado na disciplina de ASA é o *MOSS*, <http://www.cs.berkeley.edu/~aiken/moss.html>. Todos os projectos envolvidos em situações de cópia serão anulados.

### 8.4 Considerações Adicionais

Todos os programas são avaliados do modo seguinte:

```
./proj2 < entrada > saida; diff -w saida saidaEsperada
```

A impossibilidade de verificar automaticamente o resultado da execução de um dado programa implica uma penalização de 100%. Considera-se que um programa passou um teste com sucesso se o resultado produzido por esse programa for **exactamente** igual ao resultado esperado, i.e. o comando "diff" não deverá encontrar diferenças entre o vosso resultado e o esperado. Para poder ser avaliado um projecto deverá compilar correctamente num computador com o sistema operativo **GNU/Linux**, sendo o utilizado o compilador *g++* da GNU.