

Análise e Síntese de Algoritmos 2004/05

Instituto Superior Técnico

Enunciado do 1^o Projecto

Data de Entrega: 6 de Abril de 2005

1 Introdução

O 1^o projecto de ASA tem como objectivo a realização de um programa em C++ que calcula o peso da 2^a menor MST (*Minimum Spanning Tree*) de um grafo não dirigido.

2 Especificação do Problema

Seja $G = (V, E)$ um grafo não dirigido, ligado, com função de pesos $w : E \rightarrow \mathbf{R}$, tal que $|E| \geq |V|$, e tal que os pesos dos arcos são distintos. Seja \mathcal{T} o conjunto das árvores abrangentes (STs) de G e T' a MST de G . Nestas condições a 2^a menor MST é definida como a ST T tal que $w(T) = \min_{T'' \in \mathcal{T} - \{T'\}} \{w(T'')\}$. O programa a realizar deverá determinar o peso da 2^a menor MST.

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 .
- 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 peso.
- Uma linha com o carácter '#'.

- Uma lista de $N - 1$ 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 peso. Estes $N - 1$ arcos representam a MST do grafo G .

Assuma que os pesos dos arcos são distintos.

4 Dados de Saída

O programa deverá escrever o peso da 2ª menor MST.

5 Exemplos

Dados de Entrada

```
5 6
1 2 2
1 3 7
2 3 4
3 4 6
3 5 17
4 5 8
#
1 2 2
2 3 4
3 4 6
4 5 8
```

Dados de Saída

```
23
```

Dados de Entrada

```
7 11
1 2 5
1 3 6
1 4 7
2 3 1
2 5 9
3 4 2
3 6 10
4 7 11
5 6 3
6 7 4
5 7 8
#
```

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

Dados de Saída

25

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 proj1.cc.
- O compilador utilizado é o compilador gcc (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:

```
./proj1 < 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 gcc da GNU.