

Análise e Síntese de Algoritmos 2004/05

Instituto Superior Técnico

Enunciado do 3º Projecto

Data de Entrega: 3 de Junho de 2005

1 Introdução

O 3º projecto de ASA tem como objectivo a realização de um programa em C++ que resolva uma versão aproximada do problema do caixeiro viajante.

2 Especificação do Problema

Seja C um conjunto de n pontos num plano, tendo cada ponto i ($i = 1, \dots, n$) coordenadas (x_i, y_i) . Pretende-se calcular qual a sequência óptima para visitar cada um destes pontos de forma minimizar o comprimento total do caminho fechado definido por essa sequência. Este problema é conhecido pelo problema do caixeiro viajante que está provado ser NP-difícil, portanto não só não é conhecido um algoritmo eficiente para a sua resolução, como a esperança de o encontrar é pequena.

Neste trabalho considera-se o caso de se arranjar uma solução sub-óptima, considerando apenas as soluções bitónicas, o que significa que só são considerados os caminhos que podem ser percorridos indo do ponto mais à esquerda para o ponto mais à direita sempre da esquerda para a direita e vice-versa, o regresso do ponto mais à direita faz-se sempre avançando para a esquerda. Formalmente, considere-se que os índices em C correspondem à sequência bitónica óptima e que 1 é o índice do ponto mais à esquerda e j o índice do ponto mais à direita. Então tem-se:

$$\begin{cases} 1 \leq i < j, & x_i < x_{i+1} \\ j \leq i < n, & x_i > x_{i+1} \end{cases}$$

e o valor

$$\sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} + \sqrt{(x_1 - x_n)^2 + (y_1 - y_n)^2}$$

é mínimo face a todas as sequências bitónicas possíveis.

3 Dados de Entrada

O ficheiro de entrada deverá conter a informação seguinte:

- Uma linha com o número de pontos em C , n .
- Uma lista de n linhas, em que cada linha contém dois inteiros (separados por um espaço em branco) os quais representam as coordenadas x e y , por esta ordem, de cada ponto. A ordem das linhas é qualquer.

4 Dados de Saída

O programa deverá escrever o comprimento do caminho bitónico óptimo, utilizando duas casas decimais.

5 Exemplos

Dados de Entrada

```
4
0 -1
1 0
0 1
-1 0
```

Dados de Saída

```
5.66
```

Dados de Entrada

```
7
0 6
1 0
2 3
5 4
6 1
7 5
8 2
```

Dados de Saída

```
25.58
```

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 proj3.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:

```
./proj3 < 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.