

INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

2022/2023

## 2º Projecto

Data enunciado: 19 de Dezembro de 2022

Data Limite de Entrega: 06 de Janeiro de 2023 (18h00)

### Descrição do Problema

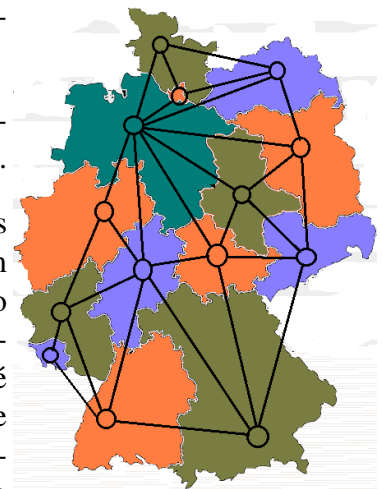
O reforço das trocas comerciais por ferrovia de alta velocidade entre regiões vizinhas da Caracolândia, foi declarada uma prioridade, dada a ineficiente situação actual por rede rodoviária. Assim, o governo da Caracolândia, decidiu encomendar um estudo preliminar ao Instituto Superior Técnico para estimar qual o valor máximo de trocas comerciais passível de ser passado para a rede ferroviária.

No entanto, algumas regiões não mantêm trocas comerciais com regiões vizinhas. Adicionalmente, devido a restrições orçamentais, o governo necessita de minimizar os custos de infraestrutura apenas financiando os troços estritamente necessários para conseguir garantir ligações entre o maior número de regiões e apenas entre as que já mantenham trocas comerciais.

De forma a simplificar o problema, podemos assumir que:

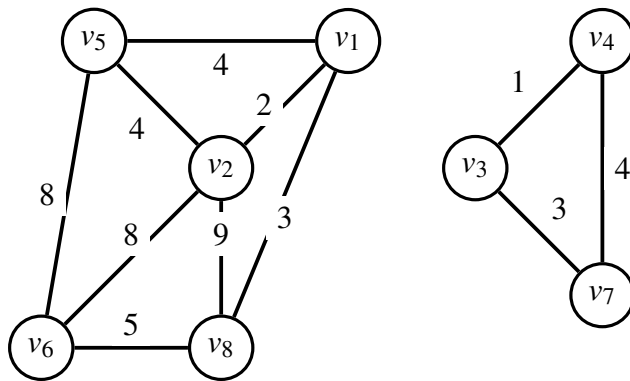
- a localização de cada região pode ser caracterizada pela latitude/longitude da estação ferroviária da sua capital;
- as ligações entre as estações ferroviárias pode ser feita em linha recta (ignorando relevo ou outros obstáculos potenciais).

Em 1934, Boris Delaunay, definiu para um conjunto de pontos  $P$  dispostos num plano, uma triangulação  $T(P)$ , tal que nenhum ponto  $p \in P$  está dentro de qualquer triângulo  $t \in T(P)$ . Usando esta triangulação de Delaunay, dado um conjunto de pontos representando as estações ferroviárias da capital de cada região, é possível construir um grafo planar não dirigido  $G = (V, E)$ , tal que  $V$  representa o conjunto de regiões e  $E$  representa as ligações entre estações ferroviárias da capital de cada região definidas como próximas pela triangulação.



Neste projecto, assumimos que: os custos de cada troço de ferrovia são os mesmos; e que após a aplicação da triangulação de Delaunay obtemos um grafo  $G = (V, E)$ , onde  $V$  designa uma região, e  $E$  o conjunto de arcos, onde a função de pesos  $w$  representa, para cada arco  $(u, v) \in E$ , o valor em M€ de trocas comerciais entre as regiões  $u$  e  $v$ .

**Problema:** dado um grafo  $G = (V, E)$  pretende-se calcular o valor máximo de trocas comerciais, minimizando os custos infraestrutura (*i.e.* número de troços).



## Input

O ficheiro de entrada contém a informação relativa a um grafo não dirigido  $G = (V, E)$ , definido da seguinte forma:

- uma linha contendo o número vértices  $|V|$  (com  $|V| \geq 1$ );
- uma linha contendo o número arcos  $|E|$  (com  $|E| \geq 0$ );
- uma sequência de  $|E|$  linhas, em que cada linha representa um arco  $(u, v)$  e contém 3 inteiros: dois inteiros que designam os vértices  $u$  e  $v$  e um inteiro que representa o valor  $w(u, v)$ .

Os identificadores dos vértices são números inteiros entre 1 e  $|V|$ .

## Output

O programa deverá escrever no output o valor máximo de trocas comerciais, minimizando os custos infraestrutura (*i.e.* número de troços).

## Exemplo

### Input 1

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

### Output 1

```
36
```

### Input 2

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

### Output 2

```
15
```

## Implementação

A implementação do projecto deverá ser feita preferencialmente usando as linguagens de programação C ou C++. Submissões nas linguagens Java/Python também serão aceites, embora fortemente desaconselhadas. Alunos que o escolham fazer devem estar cientes de que submissões em Java/Python podem não passar todos os testes mesmo implementando o algoritmo correcto.

O tempo necessário para implementar este projecto é inferior a 15 horas.

### Parâmetros de compilação:

```
C++: g++ -std=c++11 -O3 -Wall file.cpp -lm
C: gcc -std=c11 -O3 -Wall file.c -lm
Javac: javac File.java
Java: java -classpath . File
Python: python3 file.py
```

## 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.

**Relatório:** deverá ser submetido através do sistema Fénix no formato PDF com não mais de 2 páginas, fonte de 12pt, e 3cm de margem. O relatório deverá incluir uma 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. Atempadamente será divulgado um template do relatório.

**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 (<http://acp.tecnico.ulisboa.pt/~mooshak/>). Os alunos são encorajados a submeter, tão cedo quanto possível, soluções preliminares para o sistema Mooshak e para o Fénix. Note 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.

## 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 85% da nota final. Na segunda fase, o relatório será avaliado. A nota do relatório contribui com 15% 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. Os 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 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). O sistema reporta um valor entre 0 e 170.

A nota obtida na classificação automática poderá sofrer eventuais cortes caso a análise do código demonstre recurso a soluções ajustadas a inputs concretos ou outputs aleatórios/constantes.

### Deteção 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 (incluindo 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”.