

INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

## 2024/2025

### 2º Projecto

Data enunciado: 06 de Dezembro de 2024  
Data Limite de Entrega: 20 de Dezembro de 2024

## Descrição do Problema

A Câmara municipal de Caracolândia, influenciada pela nova tendência das cidades de 15 minutos<sup>1</sup>, decidiu encomendar um estudo ao Professor João Caracol, de forma a aferir a eficiência da sua rede de transporte urbano. Este estudo foca-se na análise das mudanças de linha necessárias para viajar entre quaisquer duas estações da rede de metro.

A ideia é calcular um índice de conectividade da rede metro, *metro connectivity*  $mc$ , que se define da seguinte forma:

$$mc = \max\{ML(e_1, e_2) | e_1, e_2 \in Metro\}$$

onde  $ML(e_1, e_2)$  denota o número mínimo de mudanças de linha necessárias para viajar entre as estações de metro  $e_1$  e  $e_2$ .

## Input

O ficheiro de entrada contém a informação sobre toda a rede de metro, que é definida como um multi-grafo não-dirigido, onde cada vértice representa uma estação da rede de metro e cada aresta uma secções indivisível de uma linha. Assim, o ficheiro de entrada é definido da seguinte forma:

- Uma linha contendo três inteiros: o número  $n$  de estações ( $n \geq 2$ ), o número  $m$  de ligações ( $m \geq 0$ ), e o número  $l$  de linhas ( $l \geq 0$ ) da rede de metro;
- Uma lista em que cada linha  $i$  do ficheiro contém três inteiros  $l, x$  e  $y$ , representando que a estação  $x$  está directamente ligada à estação  $y$  na linha de metro  $l$ .

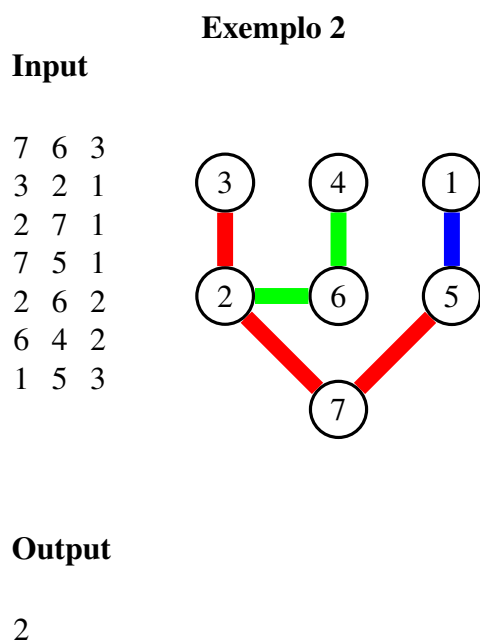
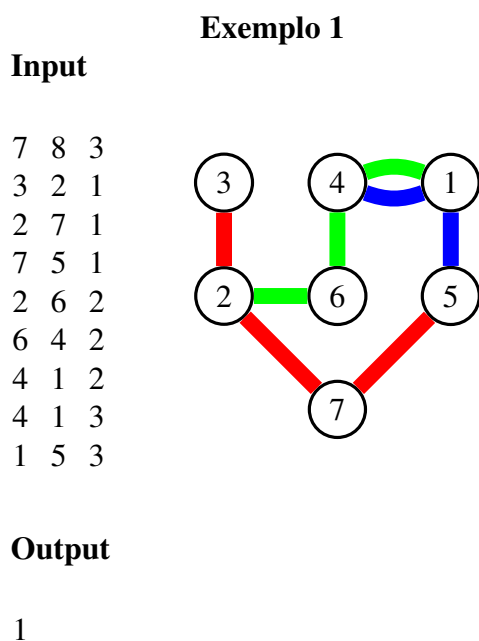
---

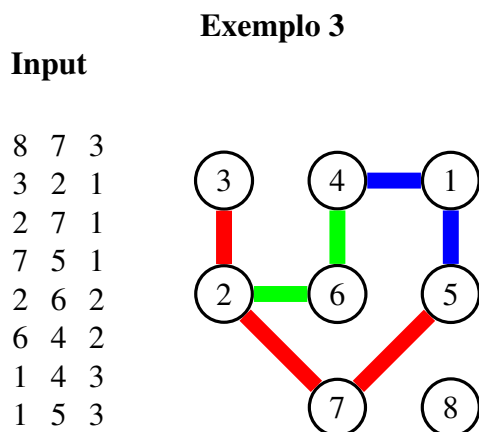
<sup>1</sup>[https://en.wikipedia.org/wiki/15-minute\\_city](https://en.wikipedia.org/wiki/15-minute_city)

Quaisquer inteiros numa linha estão separados exactamente por um espaço em branco, não contendo qualquer outro carácter, a não ser o fim de linha. Assuma que entre duas estações podem existir múltiplas conexões, correspondendo a diferentes linhas de metro.

## Output

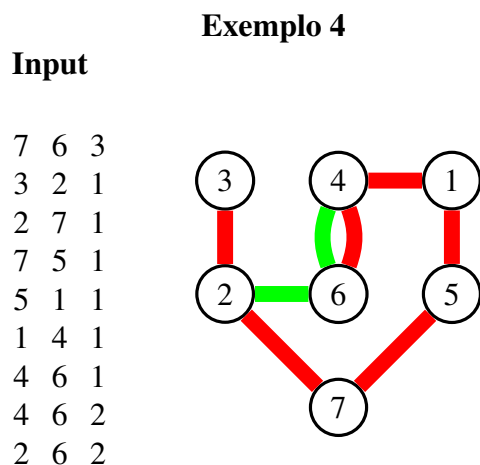
O programa deverá escrever no output um inteiro  $s$  correspondendo ao índice de conectividade da rede de metro fornecida excepto nos dois casos seguintes. Se não for necessária nenhuma mudança de linha, o output esperado é 0. Se existirem pelo menos duas estações  $e_1, e_2$  sem ligação entre si, output esperado é -1.





**Output**

-1



**Output**

0

## Implementação

A implementação do projecto deverá ser feita preferencialmente usando a linguagem de programação 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. Mais se observa que soluções recursivas podem esgotar o limite da pilha quando executadas sobre os testes de maior tamanho, pelo que se recomenda a implementação de algoritmos **iterativos**. 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 -O3 -ansi -Wall file.c -lm
Javac: javac File.java
Java: java -Xss32m -Xmx256m -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.

## **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 (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”.