

INSTITUTO SUPERIOR TÉCNICO

# Análise e Síntese de Algoritmos

## 2025/2026

### 2º Projecto

Data enunciado: 5 de Dezembro de 2025  
Data Limite de Entrega: 19 de Dezembro de 2025

## Descrição do Problema

A empresa Entregas Caracol Lda. tem uma frota de camiões que fazem entregas entre um centro de distribuição no ponto A e um centro de distribuição no ponto B (B diferente de A), numa certa área geográfica. Cada centro de distribuição corresponde a um cruzamento no mapa, existem  $N$  cruzamentos, e não existem pontes nem túneis na área de distribuição. Todas as estradas na área de entrega são de sentido único e sabe-se que não existe nenhum caminho circular que permita regressar ao ponto de onde se partiu ou a um ponto por onde se passou anteriormente.

As frequentes obras nas estradas obrigam-nos a planear cada entrega com um algoritmo que depende do número de caminhos diferentes que existem entre o ponto A e o ponto B. Os camiões são numerados entre 1 e  $M$  e, em cada dia, o número do camião que faz a entrega entre o ponto A e o ponto B é determinado calculando o número de caminhos diferentes para essa entrega, módulo  $M$ , mais um.

$$\text{NumeroCamiaoParaCaminho}(A,B) = 1 + \#\text{caminhos}(A,B) \% M$$

Dado um mapa de estradas da área, representado na forma de um grafo, a empresa Entregas Caracol Lda. quer calcular, eficientemente, quais as entregas que cada camião fará.

## Input

O ficheiro de entrada contém toda a informação sobre o mapa da área. Assim, o ficheiro de entrada é definido da seguinte forma:

- Uma linha contendo um inteiro  $N$  ( $\geq 2$ ) correspondendo ao número de cruzamentos.

- Uma linha contendo um inteiro  $M (\geq 2)$  correspondendo ao número de camiões.
- Uma linha contendo dois inteiros,  $m_1$  e  $m_2$ , que representam a gama dos números de camiões para os quais se pretendem calcular as rotas.
- Uma linha contendo um inteiro  $K (\geq 1)$  correspondendo ao número de ligações directas entre cruzamentos.
- $K$  linhas, cada uma com dois inteiros  $1 \leq a_i \leq N$  e  $1 \leq b_i \leq N$ , que representam um caminho, orientado, entre os cruzamentos  $a_i$  e  $b_i$ .

## Output

O programa deverá imprimir para cada camião, entre  $m_1$  e  $m_2$ , quais as rotas de entrega que lhe estão atribuídas. Cada linha do output inicia-se com um 'C' seguido do número do camião e um espaço, seguido por todos os pares A,B por ordem lexicográfica. Os elementos de cada par devem ser separados por uma vírgula e os pares são separados por um espaço e não devem ser incluídos os pares entre os quais não podem existir entregas.

Exemplo 1	Exemplo 2
<b>Input</b>	<b>Input</b>
5	7
3	4
1 3	2 4
8	10
1 4	1 5
1 3	1 2
1 2	1 6
4 3	2 6
4 2	2 3
3 2	2 7
2 5	3 7
3 5	3 4
	5 6
	6 7
<b>Output</b>	<b>Output</b>
C1 1,5 4,5	C2 1,2 1,3 1,4 1,5 1,7 2,3 2,4 2,6 3,4 3,7 5,6 5,7 6,7
C2 1,2 1,4 2,5 3,2 4,3	C3
C3 1,3 3,5 4,2	C4 1,6 2,7

## 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
Rust: rustc -C opt-level=3 --edition=2021 file.rs
```

## 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”.