

# Trabalho Prático I

Desenho de Algoritmos (DA) 2023

---

Catarina Canelas, 202103628

Emanuel Maia, 202107486

Diogo Geraldês, 201907847

# Leitura Dataset

Para a realização deste trabalho, foram feitas leituras de dois ficheiros csv, o “stations.csv” que representa as estações dos comboios, e o “network.csv” que representa as ligações entre as estações dos comboios.

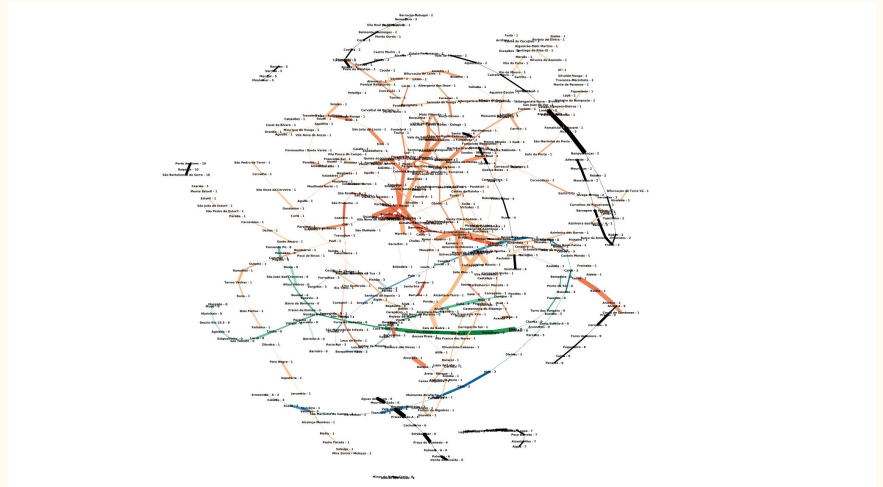
Primeiro, guardamos as estações numa tabela de hash.

Por fim, guardamos as estações como vertex no grafo criado, e as ligações entre elas como bidirecionais, adicionando ainda em cada edge o custo (que pode ser 2 ou 4) e a capacidade (encontra-se dividida por 2, pois, por exemplo, numa linha de comboios com flow de 4, são 2 comboios num sentido, e outros 2 no outro sentido).

# Grafo

O grafo tem os vertex a representar estações de comboio, e os edges a representar as ligações que existem entre essas mesmas estações.

Imagem do grafo:



## T2.1

Esta funcionalidade calcula a quantidade máxima de comboios que conseguem viajar simultaneamente entre duas estações.

São utilizados o algoritmo `findAugmentingPath` e o algoritmo `maxFlowStations`.

É calculado o max flow, que representa a quantidade máxima de comboios que conseguem viajar simultaneamente entre duas estações.

De seguida, apresentam-se um conjunto de exemplos desta funcionalidade.

## T2.1 Exemplos

Insert option:2

Insert name of origin station:Mosteiro

Insert name of destiny station:Porto Rei

1 trains can simultaneously travel between  
Mosteiro and Porto Rei

Insert option:2

Insert name of origin station:Rede

Insert name of destiny station:Alegria

1 trains can simultaneously travel between  
Rede and Alegria

Insert option:2

Insert name of origin station:Cabeda

Insert name of destiny station:Trancoso

1 trains can simultaneously travel between  
Cabeda and Trancoso

Insert option:2

Insert name of origin station:Mortagua

Insert name of destiny station:Gouveia

1 trains can simultaneously travel between  
Mortagua and Gouveia

## T2.2

Esta funcionalidade determina as estações que requerem o maior número de comboios aquando da capacidade total da network.

São utilizados o algoritmo `largestCapPair`, o algoritmo `transportationNeed` e o algoritmo `transportationNeedMun`.

De seguida, apresentam-se um conjunto de exemplos desta funcionalidade.

## T2.2 Exemplos

Insert option:3

Please wait, this may take a few minutes...

These stations require the most amount of trains when working at full capacity (11 trains):

Lisboa Oriente -> Entroncamento

Lisboa Oriente -> Santarem

Entroncamento -> Santarem

## T2.3

Esta funcionalidade calcula os top-k municípios e distritos, nos quais são precisas mais necessidades de transportes.

São utilizados o algoritmo topKMunDistr.

De seguida, apresentam-se um conjunto de exemplos desta funcionalidade.



## T2.3 Exemplos

```
Insert option:4
```

```
Insert k:5
```

```
The top 5 districts are:
```

```
SETUBAL -> 15
```

```
AVEIRO -> 13
```

```
PORTO -> 12
```

```
LEIRIA -> 12
```

```
COIMBRA -> 12
```

```
The top 5 municipalities are:
```

```
COIMBRA -> 12
```

```
LISBOA -> 12
```

```
AVEIRO -> 12
```

```
PORTO -> 11
```

```
PALMELA -> 11
```

```
Insert option:4
```

```
Insert k:8
```

```
The top 8 districts are:
```

```
SETUBAL -> 15
```

```
AVEIRO -> 13
```

```
PORTO -> 12
```

```
LEIRIA -> 12
```

```
COIMBRA -> 12
```

```
LISBOA -> 11
```

```
BRAGA -> 11
```

```
SANTAREM -> 9
```

```
The top 8 municipalities are:
```

```
COIMBRA -> 12
```

```
LISBOA -> 12
```

```
AVEIRO -> 12
```

```
PORTO -> 11
```

```
PALMELA -> 11
```

```
ESPINHO -> 11
```

```
TROFA -> 10
```

```
SANTIAGO DO CACEM -> 10
```

## T2.4

Este algoritmo encontra o número máximo de comboios que podem chegar simultaneamente a uma estação. Esta implementação cria uma super source que se conecta a todas as estações de origem (aquelas que têm apenas uma aresta na sua lista de adjacências) com capacidade infinita. De seguida retorna o max flow (usando maxFlowStations) entre a nova super source e a estação desejada.

## T2.4 Exemplos

```
Insert option:5  
Insert station name:Mortagua  
2 trains can arrive at Mortagua simultaneously
```

```
Insert option:5  
Insert station name:Cabeda  
2 trains can arrive at Cabeda simultaneously
```

```
Insert option:5  
Insert station name:Carregal do Sal  
2 trains can arrive at Carregal do Sal simultaneously
```

# T3.1

Esta funcionalidade calcula a quantidade máxima de comboios que conseguem viajar simultaneamente entre duas estações com custo mínimo para a empresa.

Consiste em encontrar o caminho mais curto entre as duas estações, recorrendo ao algoritmo de Dijkstra, calculando depois o max flow entre esses dois nós.

São utilizados o algoritmo de Dijkstra, o algoritmo `findMinResidualAlongPath`, o algoritmo `augmentFlowAlongPath`, e o próprio algoritmo `maxTrainMinCost`.

O algoritmo `maxTrainMinCost` é o que trata de calcular o custo mínimo e o max flow, implementando o algoritmo de maximum flow com custo mínimo.

O algoritmo `findMinResidualAlongPath` encontra a capacidade residual mínima no caminho encontrado pelo Dijkstra. De seguida, é chamado o algoritmo `augmentFlowAlongPath`, para aumentar o flow deste caminho com a capacidade residual mínima encontrada antes, até não haver mais nenhum caminho possível desde a origem até ao destino pretendidos.

Por fim, é calculado o custo mínimo e o max flow.

De seguida, apresentam-se um conjunto de exemplos desta funcionalidade.

## T3.1 Exemplos

```
Insert option:6  
Insert origin station:Trofa  
Insert destiny station:Esmeriz  
Numero de comboios em simultaneo (maxFlow): 2  
Custo Minimo: 10
```

```
Insert option:6  
Insert origin station:Carreira  
Insert destiny station:Carapeços  
Numero de comboios em simultaneo (maxFlow): 1  
Custo Minimo: 6
```

```
Insert option:6  
Insert origin station:Cabeda  
Insert destiny station:Trancoso  
Numero de comboios em simultaneo (maxFlow): 1  
Custo Minimo: 10
```

```
Insert option:6  
Insert origin station:Parada  
Insert destiny station:Pala  
Numero de comboios em simultaneo (maxFlow): 1  
Custo Minimo: 28
```

```
Insert option:6  
Insert origin station:Mosteiro  
Insert destiny station:Porto Rei  
Numero de comboios em simultaneo (maxFlow): 1  
Custo Minimo: 8
```

## T4.1

Este problema foi resolvido através de um grafo novo que não inclui segmentos específicos inseridos pelo utilizador. Usando este grafo, calculou-se o max flow (usando `maxFlowStations`) entre as estações desejadas.

## T4.1 Exemplos

```
Insert origin station (or q to quit): Loulé
Insert destiny station: Faro
Insert origin station (or q to quit): q
Stations for max flow:
Insert origin station: Loulé
Insert destiny station: Faro
Max flow between Loulé and Faro with reduced connectivity is 8
```

Neste exemplo, sabendo que em condições normais o max flow seria 10, podemos observar que removendo a conexão direta entre Loulé e Faro, este valor cai de 10 para 8.

## T4.2

Esta funcionalidade calcula as primeiras  $k$  estações mais afetadas por cada segmento que falha, sendo  $k$  um valor inserido pelo utilizador.

Começa por percorrer o grafo e calcular o flow normal de cada estação. De seguida é desativada a edge que liga duas estações inseridas pelo utilizador, o segmento que é pretendido falhar. A seguir é calculado novamente o flow de cada estação, mas agora com a edge pretendida desativada. Por fim, comparam-se as diferenças entre os flows e mostra-se as  $k$  estações com maior diferença, ou seja, que foram mais afetadas pela falha do segmento.

Apresentamos no slide seguinte exemplos desta funcionalidade.



## T4.2 Exemplos

Insert option:8

Insert how many affected stations will be shown

5

--- Insert the stations which path you want to disable ---

Insert name of origin station:

Santa Iria

Insert name of destiny station:

Povoa

Top 5 most affected stations.

- 1 - [Povoa] - 50.00% availability remaining - 1/2
- 2 - [Santana Cartaxo Resguardo] - 50.00% availability remaining - 1/2
- 3 - [Santana Cartaxo] - 50.00% availability remaining - 1/2
- 4 - [Reguengo - Vale da Pedra - Pontevel] - 50.00% availability remaining - 1/2
- 5 - [Virtudes] - 50.00% availability remaining - 1/2

Insert option:8

Insert how many affected stations will be shown

3

--- Insert the stations which path you want to disable ---

Insert name of origin station:

Trofa

Insert name of destiny station:

Esmeriz

Top 3 most affected stations.

- 1 - [Esmeriz] - 50.00% availability remaining - 1/2
- 2 - [Barrimau] - 66.67% availability remaining - 2/3
- 3 - [Famalicao] - 80.00% availability remaining - 4/5

Insert option:8

Insert how many affected stations will be shown

4

--- Insert the stations which path you want to disable ---

Insert name of origin station:

Louro

Insert name of destiny station:

Carreira

Top 4 most affected stations.

- 1 - [Sao Pedro da Torre] - 0.00% availability remaining - 0/1
- 2 - [Darque] - 50.00% availability remaining - 1/2
- 3 - [Areia - Darque] - 50.00% availability remaining - 1/2
- 4 - [Areosa] - 50.00% availability remaining - 1/2

# Interface com o utilizador

A interface é um menu simples para que o utilizador consiga perceber sem esforço o que esta aplicação faz.

Apresenta 8 opções, sendo a primeira para inicializar o dataset, e as restantes correspondentes a cada funcionalidade.

```
<-----> Menu <----->
```

- 1) Process dataset into database
- 2) Calculate the maximum number of trains that can simultaneously travel between two specific stations
- 3) Determine which stations require the most amount of trains when taking full advantage of the existing network capacity
- 4) Report the top-k municipalities and districts, regarding their transportation needs
- 5) Report the maximum number of trains that can simultaneously arrive at a given station, taking into consideration the entire railway grid
- 6) Calculate the maximum amount of trains that can simultaneously travel between two specific stations with minimum cost for the company
- 7) Calculate the maximum number of trains that can simultaneously travel between two specific stations in a network of reduced connectivity
- 8) Report the top-k stations most affected by each segment failure
- 0) Quit