

Trabalho prático 2

Navegação nos transportes públicos do Porto

Como certamente sabe, a **Sociedade de Transportes Colectivos do Porto (STCP)** é a empresa que gere a rede de autocarros do concelho do Porto (e também em algumas zonas limítrofes). Neste trabalho, a sua tarefa é implementar um sistema capaz de providenciar ajuda para quem quer usar a rede dos STCP para se deslocar.



Para o ajudar, terá acesso a dados reais indicando quais as paragens existentes (bem como a sua localização), quais as linhas existentes e as sequência de paragens de cada uma das linhas (em ambos os sentidos).

No final do enunciado poderá encontrar uma descrição detalhada dos dados fornecidos, que estão acessíveis a partir de um ficheiro [dataset.zip](#). A sua primeira tarefa será ler os dados fornecidos e com eles **criar um ou mais grafos sobre os quais poderá efetuar as operações pedidas**.

O sistema desenvolvido deverá ter um pequeno menu, expondo as funcionalidades que implementou de uma maneira o mais amigável possível. A tarefa mais fundamental e o requisito mínimo do trabalho é:

- **Dados dois locais, indicar o “melhor” percurso para ir de um local para o outro, usando para isso a rede dos STCP.**

Para obter pontuação máxima, o seu programa deverá ser capaz de implementar diferentes versões do que significa um “local”, do que significa “melhor” percurso, e de como pode uma pessoa mudar de linha:

- **“Local”** - tanto pode ser indicada diretamente uma paragem, como um localização (i.e., as coordenadas), onde pode ver as paragens que estejam perto dessa posição.
- **“Melhor” percurso** - um utilizador pode querer o percurso que: passa por menos paragens; percorre menor distância; implica menos mudanças de autocarro (de linha); fica mais barato (passa por menos zonas), etc.
- **“Mudar” de autocarro** – para além de mudar de autocarro numa paragem, deve também ser possível caminhar um pouco para uma outra paragem “vizinha” (que esteja a menos do que uma certa distância).

A sua implementação deverá fazer uso pelo menos uma vez de uma **pesquisa em largura (BFS)** (para grafos não pesados) e do **algoritmo de Dijkstra** para caminhos mais curtos em grafos pesados.

Se quiser incluir distâncias no seu programa, deverá usar a [fórmula de Haversine](#) para calcular a distância entre dois pontos dadas as suas latitudes e longitudes ([ver exemplo de implementação](#)).

Deverá **documentar** o seu código usando Doxygen e indicar para cada método de carácter mais algorítmico qual a **complexidade temporal** associada.

Como fatores de valorização extra, para além das diferentes variações atrás expostas e outras listagens que imagine, poderá por exemplo fazer algumas das tarefas seguintes (são apenas indicações):

- Permitir seleccionar viagens diurnas ou noturnas (*linhas com sufixo M*)
- Ilustrar o uso de algoritmos para cálculo de **minimum spanning trees (MST)** – pode por exemplo calcular a MST da rede global ou a MST de apenas uma das zonas;
- Possibilitar o fecho de paragens ou linhas seleccionadas (ex: uma zona estar com obras ou o utilizador fazer questão de não usar uma certa linha)
- Adicionar paragens e linhas de outros modos de transporte (ex: metro e/ou comboio)

Dados fornecidos para criação do sistema:

Os dados estão disponíveis em [dataset.zip](#) sob a forma de ficheiros **csv** (*comma separated values*) e foram obtidos diretamente a partir do site dos STCP. Pode criar outros ficheiros seus que facilitem uma leitura posterior do seu programa, mas nesse cada deverá incluir na sua submissão todo o código usou para transformar os ficheiros originais nos ficheiros que o seu programa lê.

- Ficheiro **stops.csv** (2487 paragens)

(primeiras 6 linhas)

```
Code,Name,Zone,Latitude,Longitude
1AL2,AV.ALIADOS,PRT1,41.14954216,-8.610023615
1AL5,ALIADOS,PRT1,41.14969203,-8.611312751
1AL6,AV.ALIADOS,PRT1,41.14914497,-8.609964267
1BS8,BOM SUCESSO,PRT1,41.15527104,-8.628342932
1CAM1,CARLOS AMARANTE,PRT3,41.17567866,-8.614147205
```

A primeira linha são os cabeçalhos. Cada uma das linhas seguintes indica o código da paragem (que é único e sem repetições), o nome “oficial” da paragem, a zona de tarifário onde está e a sua localização (latitude e longitude).

- Ficheiro **lines.csv** (73 linhas de autocarro)

(primeiras 6 linhas)

```
Code,Name
1,1 - INFANTE-PASSEIO ALEGRE
106,ZF - VALADARES (ESTAÇÃO)-FRANCELOS
107,ZC - ESTÁDIO DO DRAGÃO - AREIAS
10M,10M - AV.ALIADOS - VILA D'ESTE
11M,11M - HOSP. S. JOÃO - COIMBRÕES
```

A primeira linha são os cabeçalhos. Cada uma das linhas seguintes indica o código da linha (que é único e sem repetições) e o seu nome “oficial”. As linhas terminadas em M são linhas que funcionam apenas de madrugada (são uma espécie de rede noturna). As linhas 1 e 18 são de elétrico.

- Ficheiros **lines_[LINECODE]_[DIR].csv** (LINECODE é um código de linha, DIR é 0 ou 1)

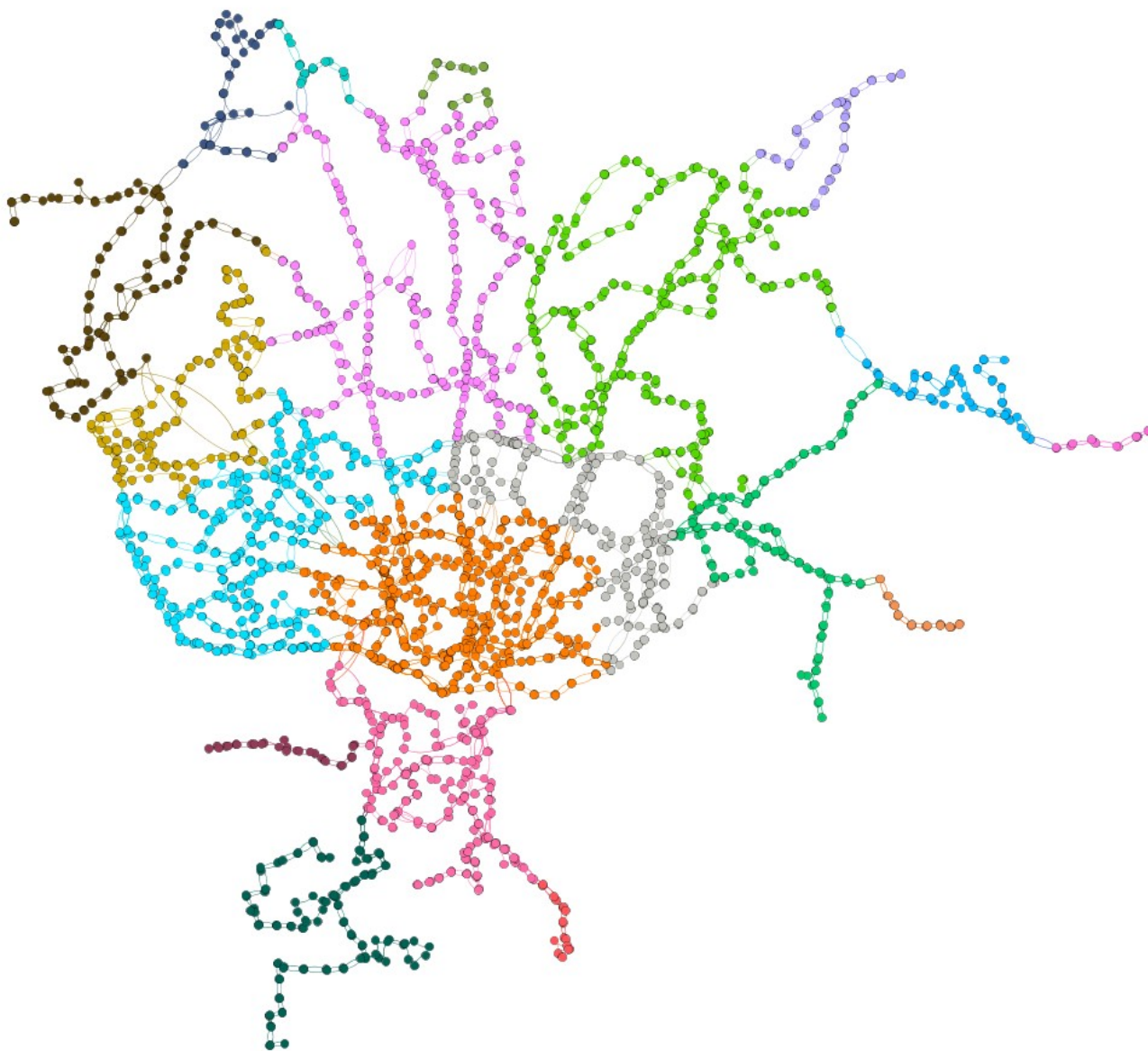
(primeiras 6 linhas de line_200_0.csv)

```
30
BLRB1
MCBL
PRDJ
PRFL
GGF
```

A primeira linha contém um inteiro n , o número de paragens. As n linhas seguintes indicam a sequência de paragens da linha, por ordem (indicando o percurso do autocarro). Neste exemplo, o 200 começa em BLRB1 (Bolhão), de lá segue para MCBL (Mercado do Bolhão), depois para PRDJ (Praça D. João I), etc. DIR indica a direção, sendo que pode ser 0 (direção “original” da linha) ou 1 (direção inversa). Existem 4 linhas circulares que só têm uma direção (300, 301, 302 e 303), pelo que a sua versão com DIR=1 contém 0 paragens.

Imagem que ilustra os dados fornecidos:

A imagem seguinte foi obtida a partir dos dados fornecidos, sendo que os nós correspondem às paragens (com a cor a indicar a zona do tarifários) e as arestas indicam duas estações sucessivas na mesma linha (não é necessário criar visualizações gráficas para o seu sistema - esta imagem é dada apenas como ilustração dos dados e da sua localização geo-espacial).



Como os dados que recebeu são reais, pode em qualquer altura verificar o resultado do seu programa (usando por exemplo o site dos STCP para ver as linhas e percursos, algo como o Google Maps ou serviço de mapas similar para ver coordenadas, etc)