Programming Project I

An Analysis Tool for Railway Network Management

Ana Almeida - up202108757
Guilherme Coutinho - up202108872
Xavier Outreiro - up202108895

Index

- Overview of the implemented classes
  - AuxMain
  - Station
  - Trip
  - Graph
- Implemented functionalities
- Used Algorithms

auxMain

- In the auxMain.h and auxMain.cpp are declared and defined functions that help in the interface with the user.
  - `void showMenu();`
  - `void showListStationAttributes();`
  - `int readOption();`
  - `string readInput();`

Station

- The station class, defined in stationTrip.h, represents the nodes of the graph;
- Its atributtes are:
  - `string name, district, municipality, township, line;`
  - `vector<Trip *> trips; // edges of the graph`
  - `bool visited = false;`
  - `Trip *path = nullptr;`
  - `vector<Trip *> incoming;`

Trip

- The Trip class, defined in stationTrip.h, represents the edges of the graph;
- Its atributtes are:
  - `Station *source;`
  - `Station *destination;`
  - `int capacity;`
  - `int flow = 0;`
  - `string service;`
  - `Trip *reverse = nullptr;`

Graph

- Graph class represents a directed graph in which the nodes are stations and the edges are trips.

- Its atributtes are:
  - `unordered_map<string, Station *> stations; // nodes of the graph`
  - `unordered_map<string, vector<Station *>> lines; // lines of the trains`
  - `int size{};`

- The function `void readFiles(const string &file1, const string &file2)` reads the given files and creates the graph.

Implemented functionalities

- List station attributes
- Shortest path between two stations
- Maximum simultaneously travelling trains between two stations
- Pairs of stations that require most amount of trains
- Top-k municipalities or districts, regarding transportation needs
- Maximum simultaneously arriving trains at a station
- Maximum trains that can simultaneously travel between two specific stations with minimum cost for the company

Used Algorithms

- All listings were made using simple for loop algorithms.
- In order to find the shortest path between two different stations we used the **Dijkstra** algorithm.
- Maximum simultaneously travelling trains between two stations, i.e. the maximum flow, was implemented using the **Edmonds-karpt** algorithm.

Used Algorithms

- The pairs of stations that require most amount of trains, top-k municipalities or districts (regarding transportation needs) and the maximum simultaneously arriving trains at a station were also calculated resorting to the **Edmonds-karpt** algorithm.
- To find the maximum number of trains that can simultaneously travel between two specific stations with minimum cost for the company we used a BFS algorithm.