Luís Alves | up202108727

2L.EIC12

DA Project

Graph Creation

```
class Edge {
    public:
        Edge(Vertex *orig, Vertex *dest, double w, const std::string &service, double cost);

        Vertex * getDest() const;
        double getWeight() const;
        double getCost() const;
        Vertex * getOrig() const;
        double getFlow() const;
        std::string getService() const;

        void setFlow(double flow);
        void setWeight(double weight);
    protected:
        Vertex * dest;
        double weight;
        double cost;
        std::string service;
        Vertex *orig;
        double flow;
};
```

- Bidirectional and weighted graph.

- Our graph reuse the code from the pratical classes.
  Each station is represented by a vertex and each connection
  between stations is represented by an edge.

Data set Reading from data files

In order to store the data, a method was implemented for each file:

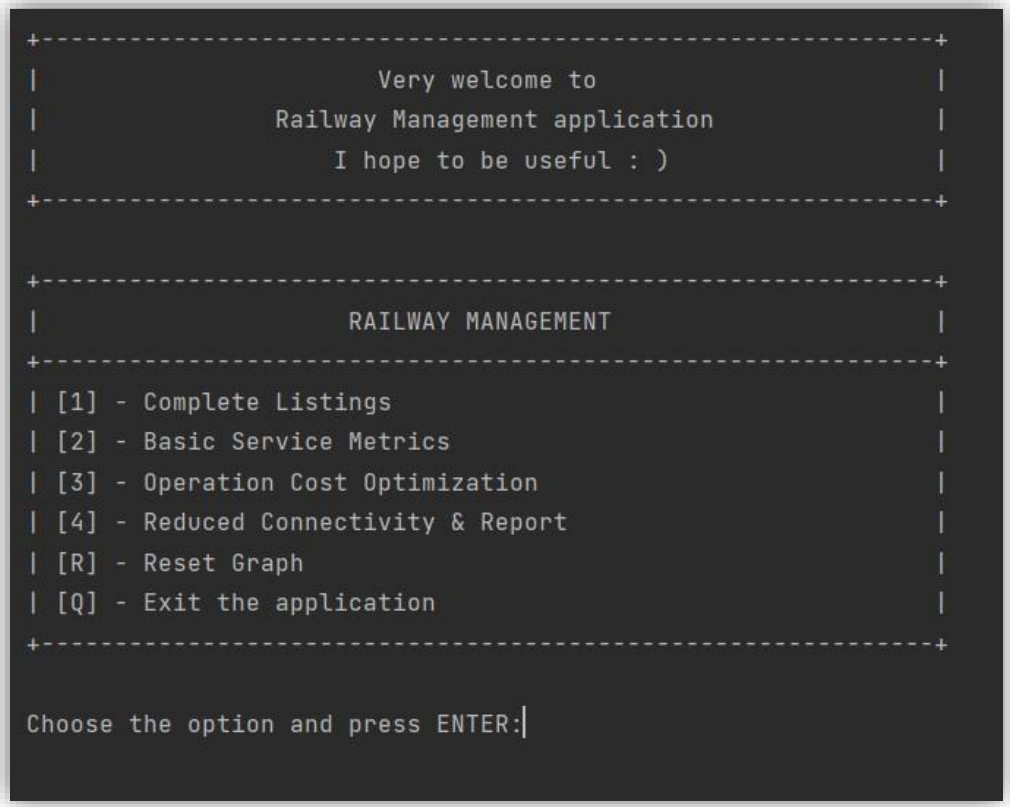
- readStations -> Reads the stations.csv file and stores each Station in a vector of Vertexs.

- readNetwork -> Reads the network.csv file and creates an Edge in the graph for each connection that is added to the respective station.

Creation of Municipalities and Districts vectors.

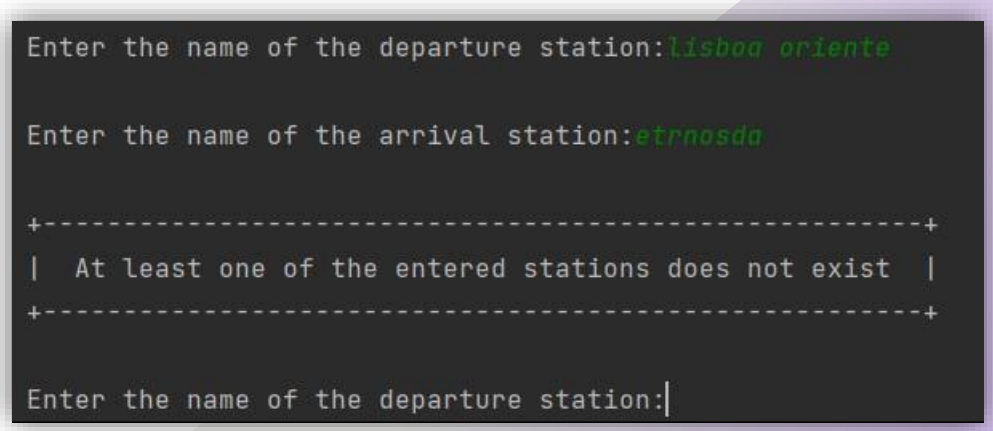
Issues with the given data.

User Interface



- This project has an interactive, intuitive and friendly menu, with multiple options chained within each different functionality.

- All user inputs are validated, displaying clear and succinct error messages when incorrect.



Menu

The menu is diveded into 6 parts:

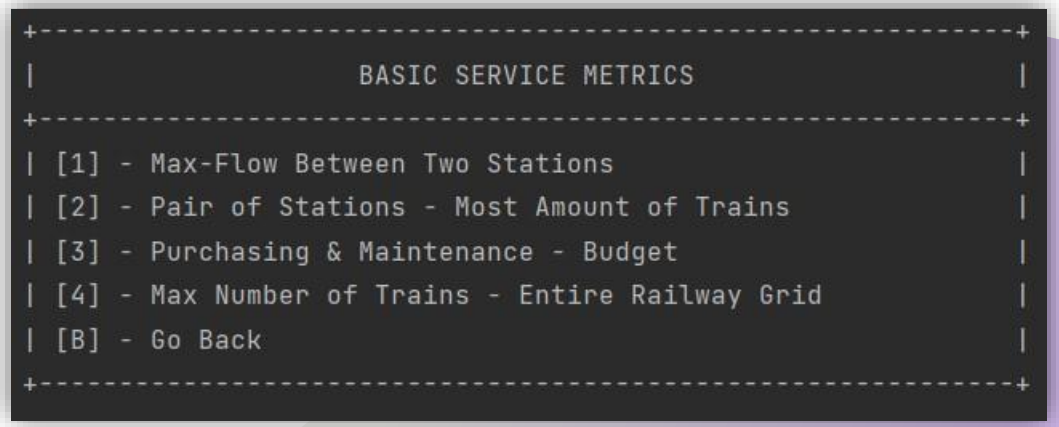
- Complete Listings
- Basic Service Metrics
- Operation Cost Optimization
- Reduced Connectivity & Report
- Reset Graph
- Exit the application

Complete Listings:

Implemented Features and Algorithms

Basic Service Metrics:

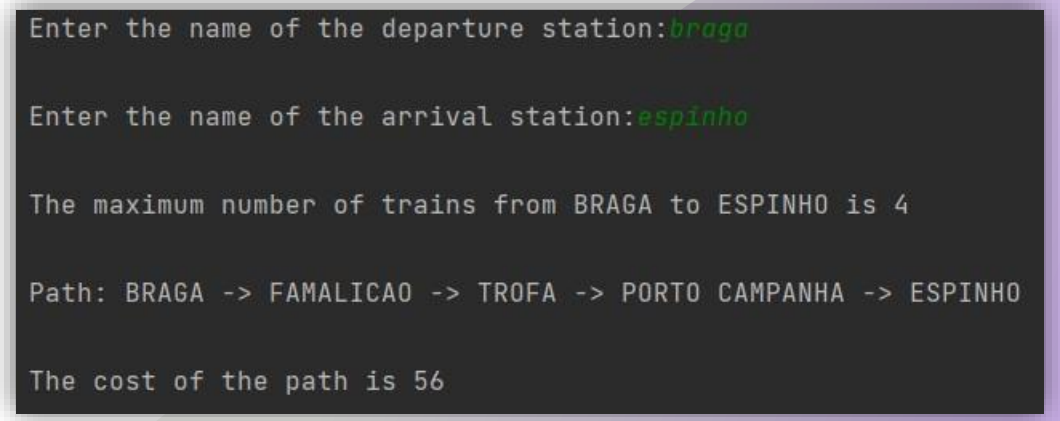
1. Edmonds-Karp
2. Max-flow between all pairs of stations
3. Maximum internal flow
4. Max-flow of the entire Railway Grid



Implemented Features and Algorithms

Operation Cost Optimization:

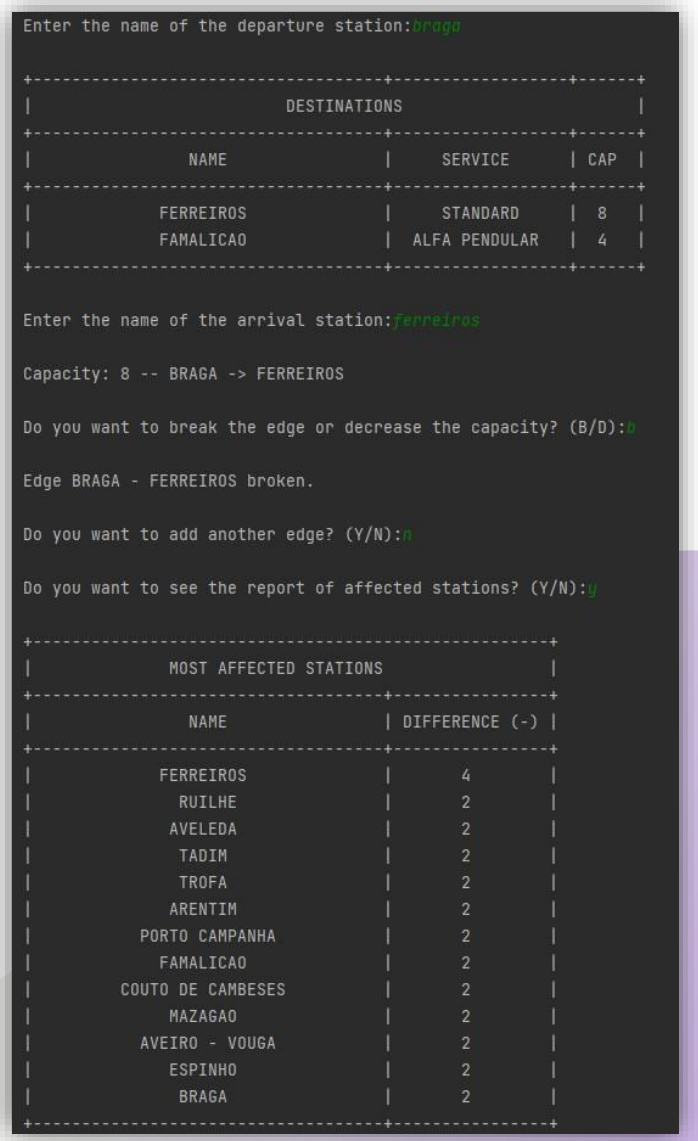
- Maximum cost of a line = capacity x service cost
- Dijkstra
- MutablePriorityQueue
- Max number of trains – Min capacity
- Calculate the path cost



Implemented Features and Algorithms

Reduced Connectivity & Report

- Break edge
- Decrease edge
- Max-flow of the entire Railway Grid
  - Before
  - After



Implemented Features and Algorithms

Reset Graph