DESIGN OF ALGORITHMS PROGRAMMING PROJECT - An Analysis Tool for Railway Network Management

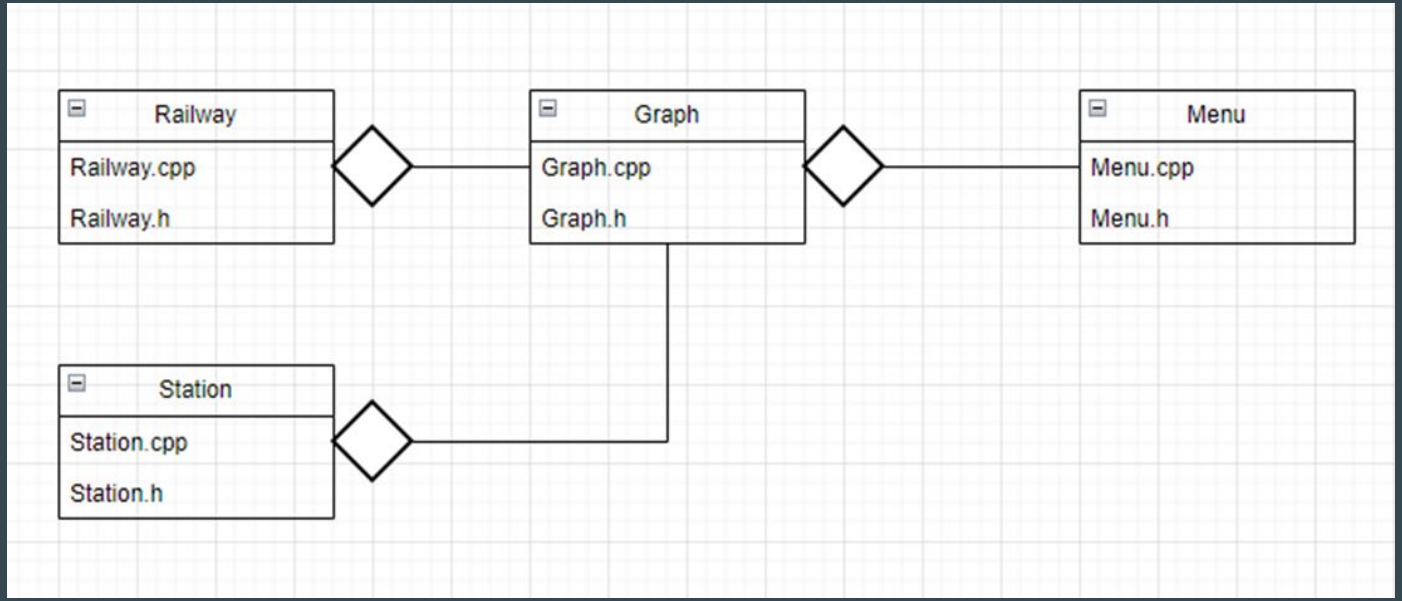
Made by group g03_2 :

•Mansur mustafin - 202102355

•José Nuno Barbosa Quintas – 202108712

•Francisco gonçalves de sousa - 202108838

Class Diagram



Dataset Reading

- When an object of the class Graph is created, the files network.csv and stations.csv are loaded into the Graph.

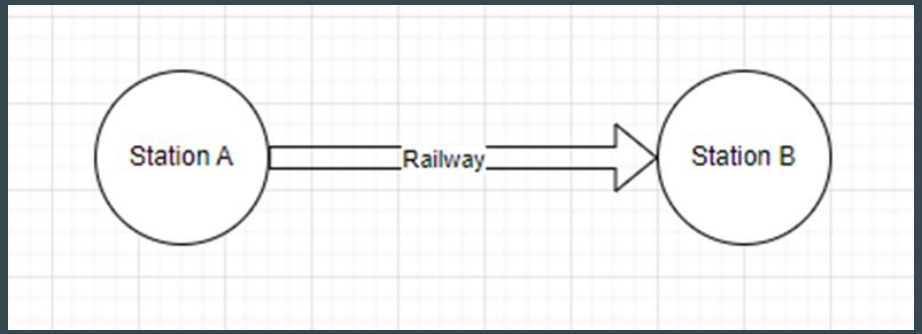
- The stations.csv file is read and the function “input_vertex” creates the objects of the class Station and stores them in a vector, defined in the Graph Class. The network.csv file is read and the function “input_edge” implements a similar process but stores objects of the class Railway in another vector, also defined in the Graph Class.

- In the class Graph, it is also defined an unordered map with an association between the name of a Station and its previously defined index.

Graph description

This is how the Graph has been organized:

- Objects of the class Station are the vertices of the Graph
- Objects of the class Railways are the edges of the Graph
- Each Station is represented by a specific integer stored in an unordered map, connecting the Station's name to this integer, named index (or Ind).



List of features and its algorithms

- Maximum flow between stations
  - Complexity: O(V1 + V2 + (E * f)). V1 – number of stations from, V2 – number of stations to, E – number of edges (or railways), f – maximum flow in the network
  - Associated algorithms – Ford-Fulkerson and DFS or Edmonds-Karp and BFS
- Railways and Pairs of Stations with the most trains
  - Complexity: O(E * ((f * V^2) + log E)). V – number of vertices (or stations), E – number of edges (or railways), f – maximum flow in the network
  - Associated algorithms – Ford-Fulkerson and DFS
- Top k Municipalities and/or Districts that require the most amount of trains
  - Complexity: O(E * (f + E)). E – number of edges (or railways), f – maximum flow in the network
  - Associated algorithms – Ford-Fulkerson and DFS

List of features and its algorithms

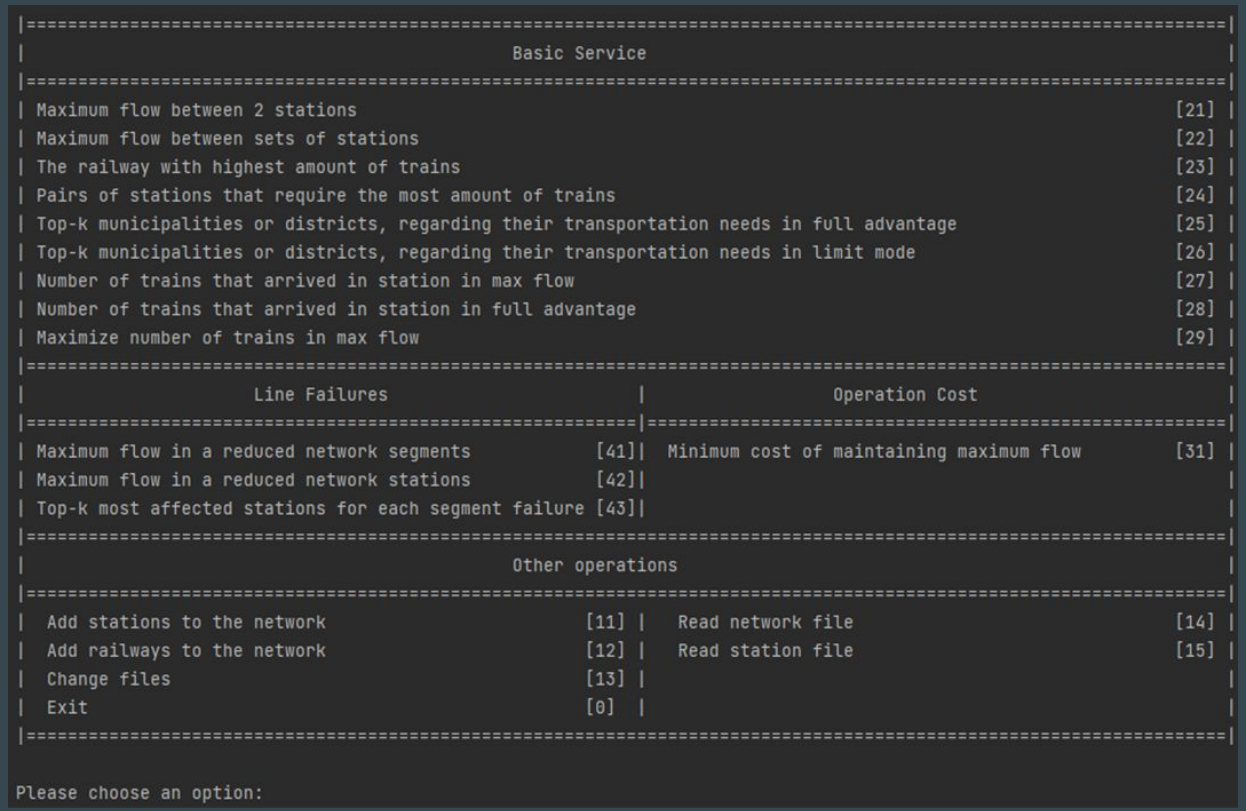
- Number of trains that arrive on a specific station
  - Complexity: O(V + (E * f)). V – number of vertices (or stations), E – number of edges (or railways), f – maximum flow in the network
  - Associated algorithms - Ford-Fulkerson and DFS
- Minimum cost to maintain maximum flow in a network
  - Complexity: O(E * (V + E) * log V). V – number of vertices (or stations), E – number of edges (or railways)
  - Associated algorithms - Dijkstra
- Maximum flow in the segments/stations of a reduced network
  - Complexity: O(S + (f * R)) . S – number of vertices (or stations) that will be added to the network (0 if we are only testing the segments of the reduced network). R – number of edges (or railways) of the reduced network
  - Associated algorithms - Ford-Fulkerson and DFS
- Most affected Stations in a reduced network
  - Complexity: O(R + f * E). R - number of elements of the reduced network (should be equal or less than E). E – number of edges (or railways) of the full network. f – maximum flow of the full network
  - Associated algorithms – Ford-Fulkerson and DFS

Other features

- Add Stations to a Graph (as vertices)
- Add Railways to a Graph (as edges)
- Change the file(s) from which the Graph’s Stations and Railways are loaded
- Read the file containing the Stations and their attributes
- Read the file containing the Railways and their attributes

User interface

Menu:



Main difficulties

The main difficulties of the project were:

- The interpretation of certain tasks, developing an organized management of the railway network
- Optimizing some of the algorithm.

Effort of each member

- Mansur Mustafin - Main Functions, Data Structure Organization
- Francisco Gonçalves de Sousa - User Interface, Menu Functions, Tests
- José Nuno Barbosa Quintas – Documentation, Presentation