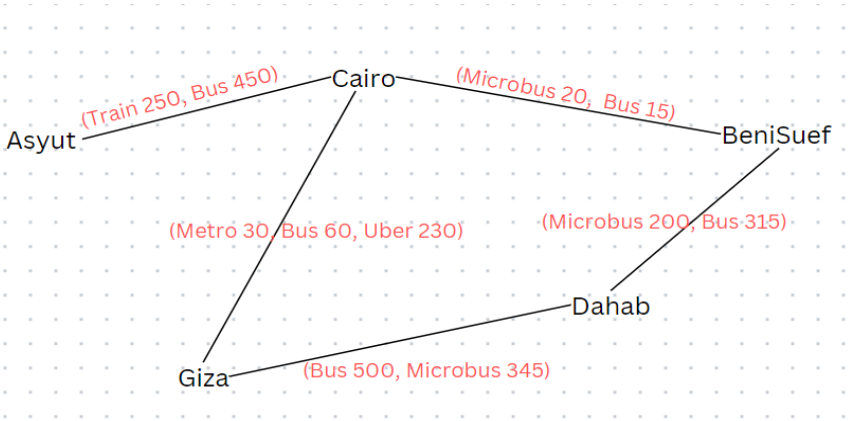
Title	Guide me
Description	You are traveling abroad, and you want to tour the country and enjoy every place in it. Based on your previous travel experience you know that there are lots of ways of transportation between cities and their prices hugely vary. You have a limited amount of money, and you want to explore the country but stay on the budget. Implement an application (*C++ program*) that reads from a file the country map (*a graph that represents the cities and all the available transportation with their prices*). The user should enter the source and destination of his trip and the amount of money he can pay, and the app should show him all the available transportation options (or a series of different transportation) that he can afford.
Minimum requirements	The program should allow the user to: 1. Traverse the transportation graph using: · Breadth First Search (BFS) · Depth First Search (DFS) The user will enter the starting node (*city name*) and choose one of the two algorithms. 2. Update/Add/Delete the transportation graph. · **Update**: Enter the source and destination and change one or more of the list of transportation. · **Add**: Enter the source and destination and add to the list of transformations (one or more). · **Delete**: Enter the source and destination and delete from the list of transformations (one or more).

	3. Determine whether the transportation map is complete or not. *(A graph in which each vertex is connected to every other vertex.)* 4. Determine two cities as source and destination and the amount of money then show a list of all the available options with mentioning the route *(one transportation method or series of them) (The route means the cities that we will pass by)* with the total cost of each option sorted ascendingly.
Building graph	The graph need to be build in the code to represent a map like this. *(List of all the available transportations per edge)* 
Bonus Opportunities	• GUI
Evaluation	• The evaluation will be mainly based on the student's ability to use and apply the most suitable data structure(s) for the given task(s) and explain why they used them and why they are better than other data structures in any given case. • Clean code is a must.

Notes	● You should **read data once at the beginning** of your run then do your operations on your structures on the code, and then save it in files at the end of your program. ● You should use the **suitable data structures** for each task *(At least 3 different data structures)* ● Using built-in libraries is allowed. ● The file structure will be sent later to the assigned teams. *(By the announcement of the projects)*
Development Tool	● C++ program.