

Project 1

Assigned: 09/04/25
Due: Monday 09/15/25 11:59 PM

Tasks:

- (1) Implement a breadth-first search and a depth-first search to find a path from Arad to Bucharest in the Romania roadmap (Figure 1). Print the nodes expanded and the frontier at each step of the search process. The node should be formatted as "*Parent_city->Current_city(Path_cost)*". When the solution is found, print the solution path and the total path cost;
- (2) Implement the A* search using the heuristic function in Fig. 2 with the same output requirements, except that the nodes should be formatted as "*Parent_city->Current_city(Path_cost, Estimated_total_cost)*".
- (3) Extra credit: collect relevant performance information and compare the time and space usage and cost optimality among the three algorithms for the Romania road map search. Provide a brief discussion of your findings.

Please use the provided Jupyter Notebook template for implementation and submit the completed Jupyter Notebook file that **includes code, documentation, and required output in your submission**. Provide clear comments to document your implementation.

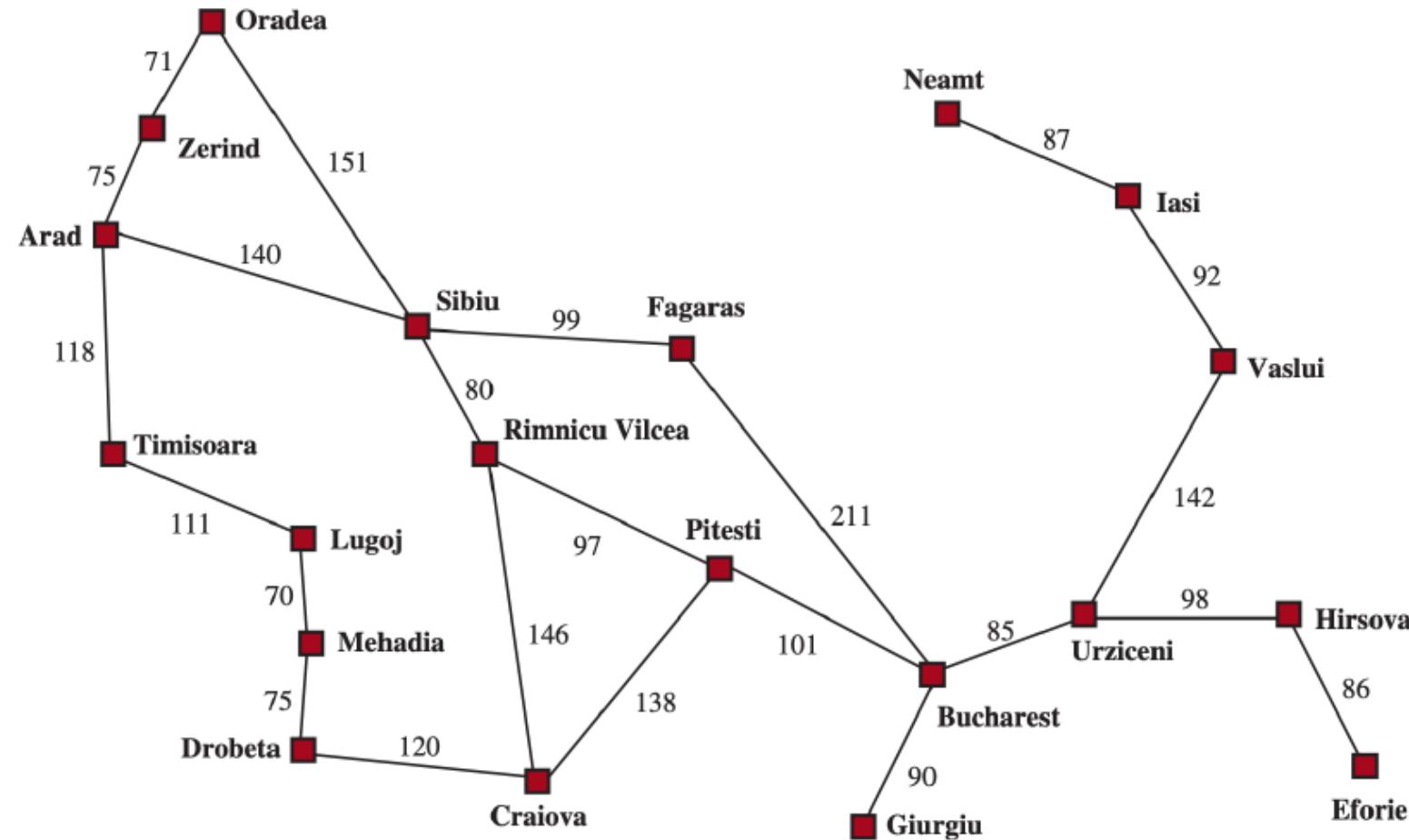


Figure 1

You may work as a group of 2 members. Please include your team member's name in your report if applicable. Otherwise, it will be treated as plagiarism if a similar code is found.

You can use utility packages, but the core search algorithm and data structure must be implemented by you.

Grading:

- 40% for the successful execution and correct output from your BFS and DFS implementation
- 40% for the successful execution and correct output for A* search implementation
- 20% for documentation, code structure, and readability
- Up to 20% extra credit for the analysis.

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Figure 2. Values of h_{SLD} —straight-line distances to Bucharest.
