

Homework Assignment #3
Assigned on October 14, 2025
Due on October 24, 2025
11:59PM on Canvas
50 points

In this assignment you are asked to implement uninformed and informed search for the Romanian road map data given in your textbook (and the figure below).

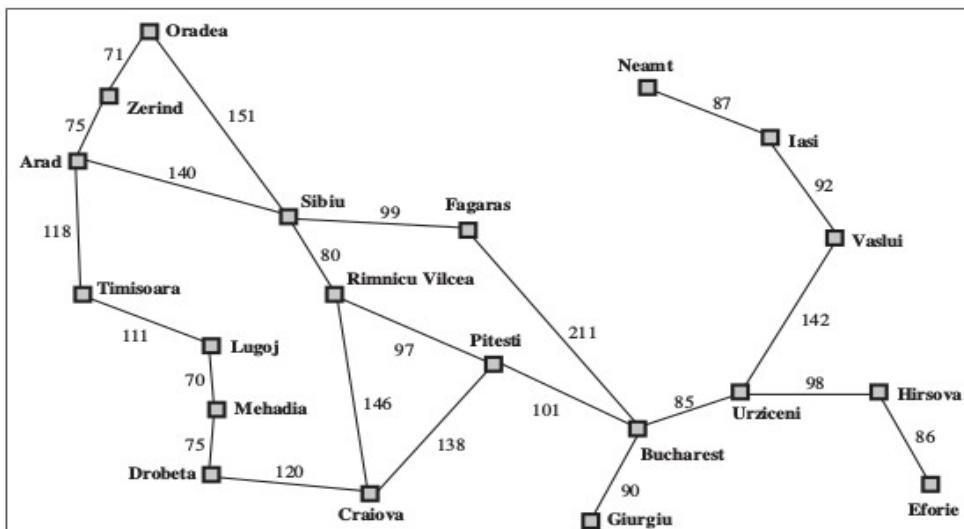


Figure 3.2 A simplified road map of part of Romania.

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

For all searches, **use Bucharest as the goal city.**

You will implement:

- (1) Breadth first (10 points)
- (2) Depth first (10 points)
- (3) A* algorithm (10 points)

Compare their performances from two points of view as follows:

(1) **Correctness (5 points)**

The algorithm finds the path from a start city to the goal city (Bucharest) or returns empty if no path exists.

(2) **Efficiency (5 points)**

Consider the **number** of cities visited before the path is found and the total cost, or the algorithm returns that there is no path (In this latter case, you may want to put a bound on the number of times a city is revisited).

Note that in each type of search **the same high-level strategy** is used as shown in the pseudocode from the textbook: maintain the nodes to be expanded in the fringe – **always implemented as a queue**. The difference is on the criterion on how the queue is constructed:

- (1) In **depth first**, the children of the current node are put in the **front** of the queue.
- (2) In **breadth first**, the list of all the children of the current node is put in the at the **back** of the queue. This will use the prolog predicate **findall**.
- (3) In the A* algorithm, the queue is maintained as a priority queue, in *nondecreasing order of the SLD, $f(n)=g(n)+h(n)$* , of the children of the current city to the goal city. In this search, the algorithm may need to backtrack to a previous node when the value of $f(n)$ is smaller than at the current node. Output the path generated and its cost. The A* algorithm should be obtained by modifying the breadth first algorithm.

Each time a node is **retrieved from the front of the queue**, the algorithm tests if it is the goal node, and when the test returns true, the algorithm ends.

To begin with, a path is empty. Eventually, the path has one node, the START node, and each node is added to it according to each search specific criterion. The path is a list of cities.

When there is no path, each algorithm should return an empty path.

What must be turned in:

1. The code, written from scratch for each search algorithm. The code must be very well commented.
2. **(10 points)** The paths (that is, the list of cities) generated by **each algorithm** from the following three cities to Bucharest: Oradea, Timisoara, Neamt. For each path, list the path cost.
3. Discussion of correctness and efficiency