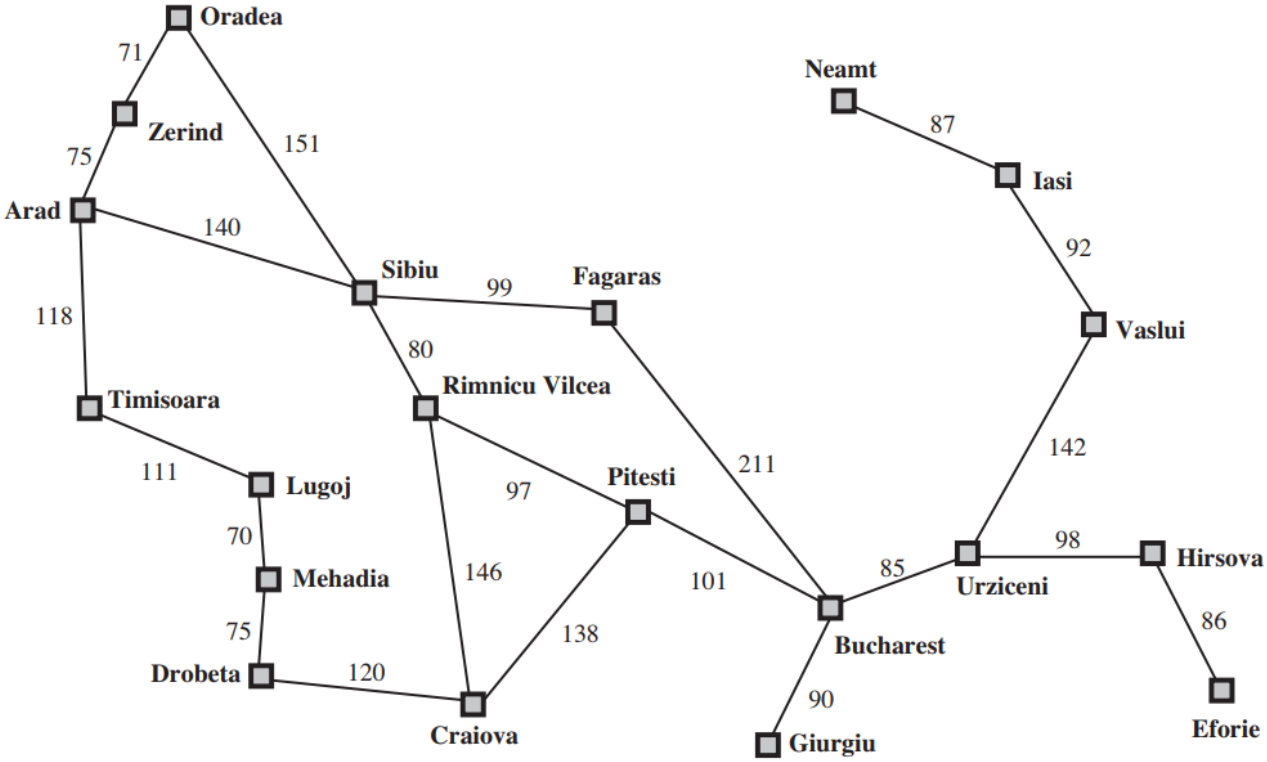
Quiz 03

Question 1.

Suppose two friends live in different cities on a map, such as the Romania map shown in below Figure



On every turn, we can simultaneously move each friend to a neighboring city on the map. The amount of time needed to move from city ***i*** to neighbor ***j*** is equal to the road distance ***d(i, j)*** between the cities, but on each turn the friend that arrives first must wait until the other one arrives (and calls the first on his/her cell phone) before the next turn can begin. We want the two friends to meet as quickly as possible.

a. Write a detailed formulation for this search problem.

b. Let ***D(i, j)*** be the straight-line distance between cities i and j. Which of the following heuristic functions are admissible?

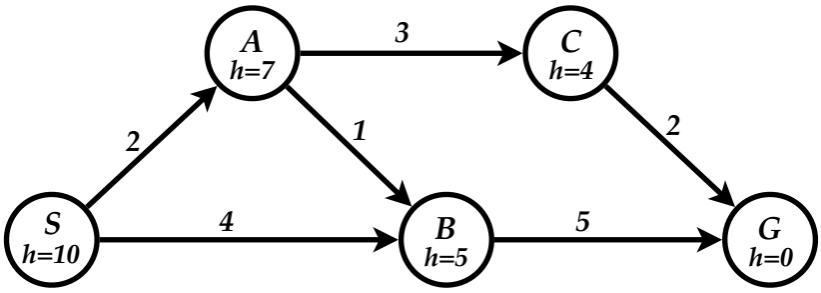
- ***D(i, j)***
- ***2 · D(i, j)***
- ***D(i, j)/2***

c. Are there completely connected maps for which no solution exists?

d. Are there any maps in which all solutions require one friend to visit the same city twice?

Question 2.

Consider the following graph, in which S and G are the initial and goal states, respectively. The heuristic values are shown under the vertices' names, while path costs are shown on every edges.



For each of the search strategies listed below,

(a) list, in order, the states expanded,

(b) list, in order, the states included in the found path, and

(c) show the final content of the frontier (recall that a state is expanded when it is removed from the frontier)

When all else is equal, nodes should be expanded in alphabetical order.

a. Breadth-first search (BFS) (shown as an example)

List of expanded nodes: S A B

Path found: S B G

Frontier = { C }

b. Uniform-cost search (UCS)

List of expanded nodes:

Path found:

Frontier = { }

c. Depth-first search (DFS) (Avoid loops by remembering nodes on the current path).

List of expanded nodes:

Path found:

d. Iterative deepening search (IDS)

List of expanded nodes for each limit: { } { } { }

Path found: .....

e. Greedy best first search (GBFS)

List of expanded nodes: .....

Path found: .....

Frontier = { }

f. A* search

List of expanded nodes: .....

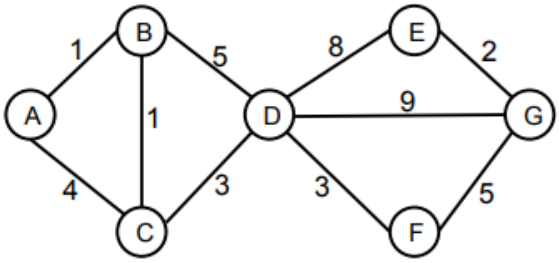
Path found: .....

Frontier = { }

g. Is the given heuristic admissible?

h. Is the given heuristic consistent?

Question 3



Node	h_1	h_2
A	9.5	10
B	9	12
C	8	10
D	7	8
E	1.5	1
F	4	4.5
G	0	0

Consider the state space graph shown above. ***A*** is the start state and ***G*** is the goal state. The costs for each edge are shown on the graph. Each edge can be traversed in both directions. Note that the heuristic ***h1*** is consistent but the heuristic ***h2*** is not consistent.

1. Possible paths returned

For each of the following graph search strategies (do not answer for tree search), mark which, if any, of the listed paths it could return. Note that for some search strategies the

specific path returned might depend on tie-breaking behavior. In any such cases, make sure to mark all paths that could be returned under some tie-breaking scheme

Search Algorithm	**A-B-D-G**	**A-C-D-G**	**A-B-C-D-F-G**
Depth first search			
Breadth first search			
Uniform cost search			
A* search with heuristic h1			
A* search with heuristic h2			

2. Heuristic function properties

Suppose you are completing the new heuristic function ***h3*** shown below. All the values are fixed except ***h3(B)***.

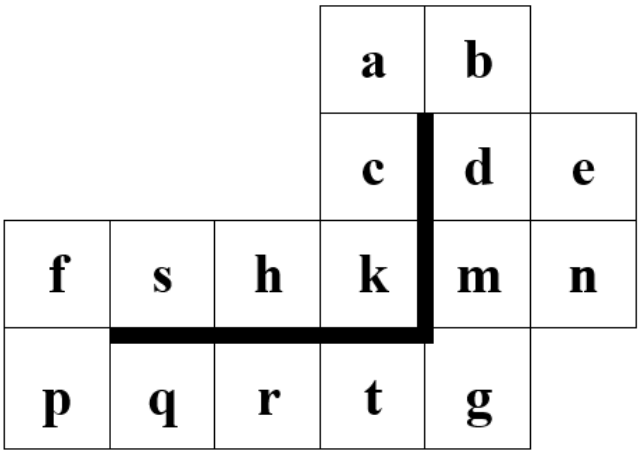
Node	**A**	**B**	**C**	**D**	**E**	**F**	**G**
h3	10	?	9	7	1.5	4.5	0

For each of the following conditions, write the set of values that are possible for ***h3(B)***.
For example, to denote all non-negative numbers, write [0, ∞], to denote the empty set, write ∅, and so on.

• What values of ***h3(B)*** make **h3** admissible?
• What values of ***h3(B)*** make **h3** consistent?
• What values of ***h3(B)*** will cause A* graph search to expand node **A**, then node **C**, then node **B**, then node **D** in order?

Question 4

Given the following maze. The bold line is wall which you cannot get pass.



Searching for the path from **s** to **g**. Give the answer with the format: **<b_1, b_2, ..., b_n>**, with b_i is explored node

a. BFS
b. DFS with a cycle check to prevent loop. The order of the operators is ***up, left, right, then down***
c. Greedy best first search with *Manhattan distances*. The Manhattan distance between two points is the distance in the x-direction plus the distance in the y-direction. It corresponds to the distance traveled along city streets arranged in a grid.
   Ex:
   $h(k) = 2$,
   $h(s) = 4$, $h(g) = 0$
d. A* with the above heuristic