



FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Marks: 100

Lab #03 + Home Task

NOTE: You must record a video while working. Your all works should be in video, if you submit the code and your activity not in video then marks will be zero. Because we have to check so many things, how you are searching on internet, how you install a new library, how you learn and follow a new library.

After completing all tasks while recording video, then upload to YouTube channel and upload your code on GitHub.

You must submit a YouTube link and GitHub link.

Task 1:

Apply BFS and DFS on tree and graph. You can use simple example for your practices.

For graph editor you can use this: https://csacademy.com/app/graph_editor/

Task 2:

1. Generate random and unique numbers of ranges 1000,40,000,80,000,200,000 and 1,000,000. So now you have 5 sets of inputs, build the tree for each sets (each set has a its own list)
2. Apply BFS and DFS on each sets and calculate the time taken for the execution to find that goal through BFS and DFS.
3. Goal should be `lis[total_len - 220]`
4. Make a data frame for your results as shown below.

Tree Size	BFS Time	DFS Time
1000		
40,000		
80,000		
200,000		
1,000,000		

5. At last plot the bar chart of the time taken by each search for example (on X-axis the data will be 1000,40,000,80,000,200,000 and 1,000,000.) and on the Y axis the data will be the seconds it took while searching.

Task 03:

For this task you must draw a tree with at least 15 nodes on your notebook and snap a picture of tree and upload to WhatsApp group. First come, first served will apply to this situation.

Before constructing the tree, each student must ensure that his or her group is not constructing the identical tree as discussed in the lab. Because your time will be monitored, you must first upload the image and then begin coding.

Tree Size: minimum 15 nodes and you can increase the size.

Visualize the tree by following libraries.

1. Binarytree <https://pypi.org/project/binarytree/>
2. Anytree (build both tree shapes as mentioned in link)
<https://anytree.readthedocs.io/en/latest/>
3. Treelib <https://treelib.readthedocs.io/en/latest/>
4. Turtle (For Animation)
5. Graphviz (install using pip also you have to download the software)

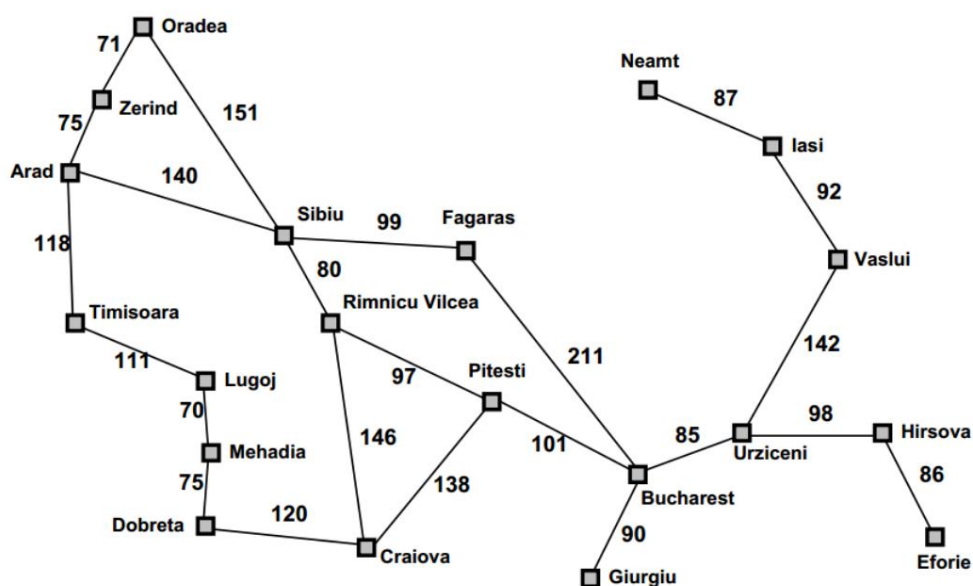
Task 4:

1. Apply Uniform Cost Search (UCS) on simple example.
2. Now you must create a code that accepts the graph value created in the graph editor as described in task 1. And converted to dictionary format, as we did in the lab. Review the UCS code for assistance.
3. Apply UCS on Arad Graph (Check book for Arad Map)
4. Also you have to find the exact path, there are many path to node "e" or "g". So, what is the minimum path with nodes.

Task 5:

Visualize the Task 4 part 1 graph using networkx library.

Visualize the Arad map graph (impressive designing should be high value, your graph should be exact like Arad, your graph should not be move, Arad node should be in same place as shown in book)



Task 6:

Apply BFS and UCS on N-Queen problem.