

LE - 7

Graphs

Introduction:

The lab exercise LE - 7 is a hands-on component for the concepts covered in the lectures on Graphs.

Expected Functionality:

The following are the expectations associated with LE - 7:

1. Implement the **addNode()** method to add a new node to the Graph dictionary. New nodes are initiated without any edges attached i.e., an empty adjacency list. This takes in the node value to be added.
2. Implement the **addEdge()** method create an edge between the two nodes and add the edge to the adjacency list of each node along with its associated weight. This takes in the source node (src), the destination node (dest) and the weight to create the edge.
3. Implement the **removeNode()** method to remove a node from the graph along with its adjacency list and any associated edges connected to other nodes from the respective adjacency lists. This takes in the node value to be removed.
4. Implement the **removeEdge()** method to remove an edge connecting two nodes. This takes in the source node (src) and destination node (dest) for which the connecting edge has to be deleted. *Note: this should delete the edge from the adjacency lists of both nodes.*
5. Implement the **degree()** method to return the degree of a node i.e., the number of connections it has with other nodes. This takes in the node value for which to return the degree.

Additionally, explore the **NewtorkX** library that is used to create and visualize graphs. Use your output graph dictionary containing the adjacency lists to initiate a graph structure with the relevant NetworkX method and create a visualization of the graph. The more information the visualization contains the more points you get out of 5.

Starter Code:

You are provided with the *LE7_Graphs.ipynb* file. The notebook contains a template class for Graph. Add the methods needed for implementing the aforementioned functionality. The following methods are part of the class:

__init__(V): initiates a graph dictionary with V number of vertices.

addNode(node): adds a new node to the graph with no connected edges.

addEdge(src, dest, weight): adds an edge to an undirected graph.

removeNode(node): remove the node from the graph by deleting the key-value pair.

removeEdge(src, dest): deletes an edge between the src and dest nodes from the adjacency list.

degree(node): returns the degree of a node i.e., how many other nodes it is connected to.

getVertexCount(): returns the number of vertices or nodes in a graph.

printGraph(): prints the graph in the format edge: adjacency list.

Rubric:

Your code will be tested with provided test cases.

Location of the code:

The code would be provided at:

https://colab.research.google.com/drive/1XXKMeX6ScOe0LFF79I4mWj78izvEG_bs?usp=sharing

What to do when done:

Once you have completed the exercise, you should upload it to the codePost (will be visible soon).

Please ensure the following while submitting:

1. Once satisfied with your code, you should download the file as a python script (.py file),
2. by going to **File > Download > Download .py**
3. The name of the file should be LE7.py
4. You can run the test cases on your script up to a limit of 50 times.
5. Once satisfied with the test runs, complete your submission.