Experiment no 9

Graphs Implementation

OBJECTIVES:

- To learn about the concept of Graphs.
- To learn different ways for implementing a Graphs.

1. GRAPHS IMPLEMENTATION:

A graph is a pictorial representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented by points termed as vertices, and the links that connect the vertices are called edges. The various terms and functionalities associated with a graph is described in great detail in our tutorial here.
In this chapter we are going to see how to create a graph and add various data elements to it using a python program. Following are the basic operations we perform on graphs.

- Display graph vertices
- Display graph edges
- Add a vertex
- Add an edge
- Creating a graph

A graph can be easily presented using the python dictionary data types. We represent the vertices as the keys of the dictionary and the connection between the vertices also called edges as the values in the dictionary.
Take a look at the following graph:

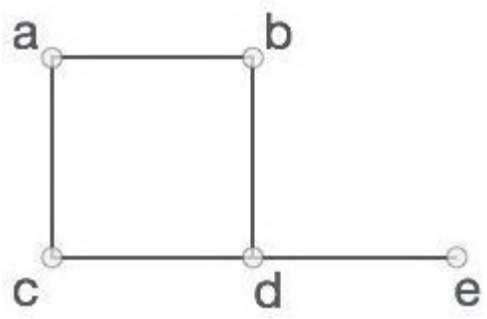


Figure 1: Graph Implementation

In the above graph,
V = {a, b, c, d, e}
E = {ab, ac, bd, cd, de}

```
# Create the dictionary with graph elements
graph = {
   "a" : ["b","c"],
   "b" : ["a", "d"],
   "c" : ["a", "d"],
   "d" : ["e"],
   "e" : ["d"]
}
# Print the graph
print(graph)
```

Figure 2: Simple Example of Graph using dictionaries

2. DISPLAY GRAPH VERTICES:

To display the graph vertices we simple find the keys of the graph dictionary. We use the keys() method.

```
class graph:
   def __init__(self,gdict=None):
      if gdict is None:
         gdict = []
      self.gdict = gdict
# Get the keys of the dictionary
   def getVertices(self):
      return list(self.gdict.keys())
# Create the dictionary with graph elements
graph_elements = {
   "a" : ["b","c"],
   "b" : ["a", "d"],
   "c" : ["a", "d"],
   "d" : ["e"],
   "e" : ["d"]
}
g = graph(graph_elements)
print(g.getVertices())
```

Figure 3: Graph Vertices

3. DISPLAY GRAPH EDGES:

Finding the graph edges is little tricker than the vertices as we have to find each of the pairs of vertices which have an edge in between them. So we create an empty list of edges then iterate through the edge values associated with each of the vertices. A list is formed containing the distinct group of edges found from the vertices.

```
class graph:
   def __init__(self,gdict=None):
      if gdict is None:
         gdict = {}
      self.gdict = gdict

   def edges(self):
      return self.findedges()
# Find the distinct list of edges
   def findedges(self):
      edgename = []
      for vrtx in self.gdict:
         for nxtvrtx in self.gdict[vrtx]:
            if {nxtvrtx, vrtx} not in edgename:
               edgename.append({vrtx, nxtvrtx})
      return edgename
# Create the dictionary with graph elements
graph_elements = {
   "a" : ["b","c"],
   "b" : ["a", "d"],
   "c" : ["a", "d"],
   "d" : ["e"],
   "e" : ["d"]
}
g = graph(graph_elements)
print(g.edges())
```

Figure 4: Graph Edges

4. ADDING A VERTEX:

Adding a vertex is straight forward where we add another additional key to the graph dictionary.

```
class graph:
   def __init__(self,gdict=None):
      if gdict is None:
         gdict = {}
      self.gdict = gdict
   def getVertices(self):
      return list(self.gdict.keys())
# Add the vertex as a key
   def addVertex(self, vrtx):
      if vrtx not in self.gdict:
         self.gdict[vrtx] = []
# Create the dictionary with graph elements
graph_elements = {
   "a" : ["b","c"],
   "b" : ["a", "d"],
   "c" : ["a", "d"],
   "d" : ["e"],
   "e" : ["d"]
}
g = graph(graph_elements)
g.addVertex("f")
print(g.getVertices())
```

Figure 5: Adding a vertex

5. ACTIVITIES:

A1) Write a code for Graph implementation as given in the Graph Implementation topic in manual.

A2) Write a code for inserting vertices and edges in a graph.

A3) Write a code for inserting a vertex in a graph.

IMPORTANT: All the activities` code should be attached to the manual before summary section.

Lab Exercise and Summary

Summary should cover Introduction, Procedure, Data Analysis and Evaluation.

Student's Signature:_____ **Date:**_____

LABORATORY SKILLS ASSESSMENT (Psychomotor)

Total Marks:100

Criteria (Max Marks)	Level 1 0% ≤ S < 50%	Level 2 50% ≤ S< 70%	Level 3 70% ≤ S< 90%	Level 4 90%≤ S ≤100%	Score (S)
Procedural Awareness (30)	Selects inappropriate skills and/or strategies Required by the task.	Selects and applies appropriate skills and/or strategies required by the task with major errors.	Selects and applies the appropriate strategies and/or skills specific to the task without significant errors.	Selects and applies appropriate strategies and/or skills specific to the task without any error.	
Practical Implementation (30)	Makes major critical errors in applying procedural knowledge related to python Graphs Implementation	Makes numerous critical errors in applying procedural knowledge related to python Graphs Implementation	Makes some non-critical errors in applying procedural knowledge related to python Graphs Implementation	Applies the procedural knowledge in optimized ways related to python Lists, Graphs Implementation	
Use of Tool/Equipment (30)	Uses tools, equipment and materials with limited competence.	Uses tools, equipment and materials with some competence.	Uses tools, equipment and materials with considerable competence.	Uses tools, equipment and materials with a high degree of competence.	
Safety (10)	Requires constant reminders to follow safety procedures.	Requires some reminders to follow safety procedures.	Follows safety procedures with only minimal reminders.	Routinely follows safety procedures.	
				Marks Obtained	

Instructor Name:_____ **Sign:** _____

LABORATORY SKILLS ASSESSMENT (Affective)

Total Marks: 40

Criteria (Max. Marks)	Level 1 0% ≤ S < 50%	Level 2 50% ≤ S < 70%	Level 3 70% ≤ S < 90%	Level 4 90% ≤ S ≤ 100%	Score
Introduction (5)	Very little background information provided or information is incorrect	Introduction is brief with some minor mistakes	Introduction is nearly complete, missing some minor points	Introduction complete and well-written; provides all necessary background principles for the experiment	
Procedure (5)	Many stages of the procedure are not entered on the lab report.	Many stages of the procedure are entered on the lab report.	The procedure could be more efficiently designed but most stages of the procedure are entered on the lab report.	The procedure is well designed and all stages of the procedure are entered on the lab report.	
Data Record (10)	Data is brief and missing significant pieces of information.	Data provides some significant information and has few critical mistakes.	Data is almost complete but has some minor mistakes.	Data is complete and relevant. Tables with units are provided. Graphs are labeled. All questions are answered correctly.	
Data Analysis (10)	Data is presented in very unclear manner.	Data is presented in ways that are not clear enough.	Data is presented in ways that can be understood and interpreted.	Data is presented in ways that best facilitate understanding and interpretation.	
Report Quality (10)	Report contains many errors.	Report is somewhat organized with some spelling or grammatical errors.	Report is well organized and cohesive but contains some grammatical errors.	Report is well organized and cohesive and contains no grammatical errors. Presentation seems polished.	
				Marks Obtained	

LABORATORY SKILLS ASSESSMENT (Cognitive)

Total Marks: 10

(If any) **Marks Obtained**	

Instructor's Signature:_____ **Date:**_____