**Data Structures and Algorithms**

**Lab 13**

**Submitted To:**

Mr. Dilshad Sabir

**Submitted By:**

Manaal Waseem

FA18-BCE-074

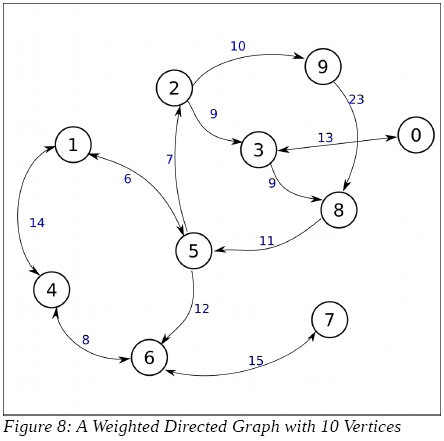
**In Lab:**

**Task 1:**

***Complete the Adjacency Matrix given as ‘my\_graph[][]’ in the main function of the skeleton code provided. You will use the following figure for completing this matrix. Call the function:***

***‘add\_edge()’***

***with correct weights to fill in the matrix.***



**Program:** Code is as follows:



**Output:**



**Task 2:**

***For this part you will have to complete the function:***

***‘find\_path\_dfs()’***

***which finds the cost of going from the source vertex (src) to destination vertex (dst) using Depth First Search (DFS) algorithm. You will use a stack for implementing DFS algorithm.***

**Program:** Code is as follows:

****

**Output:**

****

**Post Lab:**

***Complete the code for Task 2 and submit along with a report explaining your implementation.***

**Explanation :**

I have completed the code by following the DFS algorithm. Firstly the node that is currently being visited is pushed into the stack along with the path cost. Afterwards, top of the stack is popped and it is assigned to **crnt\_exploring** and its cost is assigned to **path\_cost**. Then a **for loop** iterates from **0** upto **size**. Inside this loop all the neighboring nodes of **crnt\_exploring** node that are unvisited are pushed into the stack.

Once this exploration is completed, the stack is popped to select the next node to visit and it is assigned to **crnt\_visiting**. The **path\_cost** is calculated by adding the **cost\_to\_visit** of newly selected node in **path\_cost**.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THE END**