

Name: Ishraq, Kanad Adib
Section: 20

Assignment 4 | ID: 22299325

Task 1(a)

Firstly I have taken input edges on an array and then I created zero matrix using the dimension of the vertices and then ~~appended the~~ put the weight of ~~each~~ node.

Task 1(b)

created an adjacency dictionary and appended the ~~edges~~ nodes and their connection on a bidirectional graph

Task 2

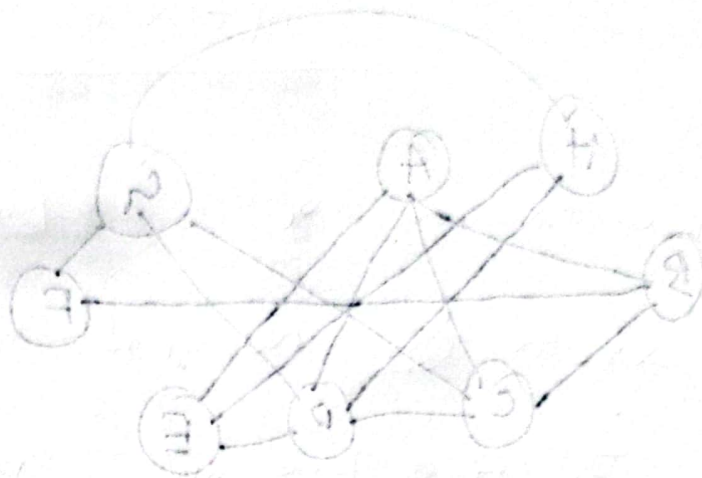
Firstly created a graph, then imported a library to create a queue and starting node is 1. and kept an state array to know the current state then in iterative approach we got our BFS result.

Task(3)

Firstly, created an adjacency list and used it as our graph and then using ~~stack~~ stack and iteration and state array we got our DFS answer.

Task(4)

If we modify our previous code slightly of DFS and keep a flag to determine we can easily get our cycle or not cycle answer and we used iterative approach.



Task (5)

We will use our BFS code from task 2 and then we will keep an parent array to know the parent of the path and then we will run while loop to get our shortest path until it reaches the start and then reverse the answer.

Task (6)

Task (6)

We will use two functions; one for flood fill and one ~~for~~ to count the max diamond. In the flood fill function we will put the ^{corner} ~~edge~~ cases and some special cases as our recursive stopping condition and we will use recursion to count up, down, left, right of the matrix.