Here we have er?

Task 1(a)

Here we need to create an adjacent matrix. As the graph is directed so we took all the output ~~ab~~ and we replaced the row column withe weight.

Task - 2 (b)

Here we used adjacent ~~mat~~ list to ~~find~~ ~~and~~ ~~the~~ get all of our values. We used dictionary here to store the values ~~a~~ ~~our~~ according to their correspondine weigh

Task 2

I have created an object of every number possible in the graph. All the objects has the same color poperly and vertex property to store their verteciles. In bfs traversal we traverse all the verteciles first, in other words all the levels first.

Task - 3

Here we I used stack implementation to do our dfs traversal. We found one vertex and went depth of it to find all the corresponding vertecies.

Task - 4

We used the DFS traversal to find a cycle If we have found it with a color value 1 again then

We are sure that the graph has a cycle. Else, the dfs function returns None which means no cycle.

task-5

Here I have used a the bfs traversal method to traverse the whole graph. Additionally I made an array with minus one. I updated the array changing the element number th of index to the poped / the present value to track the parent. Then traverse that one to find shortest path.

task-6

Here we created a 2D matrix and traverse the whole thing. Then we used an exception. Dfs traversal to track our available path and find the total number of Diamon Then we return the value. Then we check if we found any better number of diamond while traversing. Note if I already traversed the whole array we build a wall so that in the next traversal we don't use the same path.