Theme:

Date: / /
☐Sat ☐Sun ☐Mon ☐Tue ☐wed ☐Thu ☐Fri

Task 1.a — After using list comprehension for making the base, I used a for loop to add weighted value into the vertices places in the list. and finally printed using join() and a for loop.

Task 1.b — Just like Task 1.a, I used list comprehension to make adjacency list. then in ~~the same way~~ I stored r_c in a tuple then used a for loop with join() to print the output.

Task 02 Here we used Queue structure to build BFS. If a node is visited, it will be 1 from 0, and printed the sequence

Task 03 Same as Task 2. But used stack data structure to establish DFS and traverse the input and print the visited nodes sequence as output.

Date: / /
☐Sat ☐Sun ☐Mon ☐Tue ☐wed ☐Thu ☐Fri

Task 04 used DFS to find if there's any cycle. If a cycle is present, its print yes otherwise, No. there I used 2 functions to keep track of the nodes and compare them.

Task 05 : Here we used the queue data structure to generate the BFS in order to find the shortest path.

Task 06 there we used DFS to find the longest path so that we can collect more Diamonds.