Task 1 (A)

We make a 2d array containing 0 as elements and iterate this 2d array using input's row and column and values to make the matrix.

Task 1 (B)

We make a 2d array of empty lists and then using the inputs row values, append in those empty lists which node the row value's node will lead to and the value of edge as tuple.

Task 2

We first make a graph dictionary using the inputs.. And then we use BFS function to do BFS traversal we use colour list to keep track of visited nodes and use q list to append the element we're visiting and pop the first element of q list.

Task 3

We make a graph dictionary again, create colour list to keep track of visited nodes and use DFS function recursively to complete DFS traversal.

Task 4

In this task, we use DFS to solve this problem. Lines is a list of directed edges. Mapcycle function checks visited nodes that sends edges into cycle function, which with the help of DFS traversal determines if there's a cycle in the graph or not.

Task 5

We use BFS traversal to find the shortest path. Queue class helps doing BFS, Vertex class through objects keeps track of vertices, their parent node, adjacent Vertices and distance from source. We just check the shortest distance for d after BFS traversa

Task 6

We use flood fill algorithm using DFS to find out the maximum number of diamonds. flood fill function applies DFS to all non obstacles and stones the diamond counts in an array and finally we return the maximum number of diamonds found.