

Ans to the ques no: 4

1 For adjacency list (which I have used for task 2 and 3), time complexity will be,

$$O(V+E).$$

Here,

V = number of vertices

E = " " edges

In BFS, the first while loop will run until the queue is empty. and it will happen when all the vertex is visited. And, for loop will run for every edges of that vertex. So, the running (time) $O(\text{vertices} + \text{edges})$

gf, I used matrix then it would be $O(V^2)$. because it will ~~be~~ iterate all the $[i][j]$ th element (even if the $[i][j]$ element is 0).

For DFS, we identify all of its
neighbour nodes by crossing its adjacency
list. Here, also, each vertex and edge
is visited once, so, the time complexity is
also $O(V+E)$.

also,
if we use matrix then for this time complexity
would be $O(V^2)$, same as BFS.

Output (BFS)

1 2 3 4 5 7 11 6 12

Output (DFS)

1 2 3 4 7 11 12 6

DFS

For, by traversing, Gary will get to
the victory road first. Although, we

Know that, to find the shortest path by BFS. Because it ensures every ^{nodes of a} level in graph then go to the next level. whereas in DFS, it goes depth of a graph and then return to previous level. So, ~~in~~ to be at safe side BFS can give us shortest path. though here, Gary can get to the victory road ^{first.} by traversing DFS