Task-05

Now for task-2 where we used bfs the time complexity will be:

```
def BFS (visited, graph, node, endpoint)
        Do visited [int(node)-1] = 1
        Do queue.=append (node)
        while queue not empty
         Do m = pop()
         print m
         if m = endpoint break
         For each neighbour of m in graph
           if visited [int(neighbour-1] = 1
              Do visited [int (neighbour-1) = 1
              Do queue.append(neighbour)
```

In the case of adjacency list the first while loop will go for traverse in the node which will take O(V) time and in the case of for loop it will be going through its neighbours of node and this will take O(E) time. So, total time = O(V+E)

In the case of adjacency matrix every node and all of its edges needs to be traversed and a matrix have same row and column so it will take time of $O(V^2)$.

Now in Task 3 for DFS

```
DFS_VISIT(graph, node)
    Do visited[int(node)-1] = 1
    Printed.append(node)
    For each node in graph[node]
        if node not visited
            DFS_VISIT(graph, node)

DFS(graph, endpoint):
    For each node in graph
        if node not visited
            DFS_VISIT(graph, node)
```

Print "Printed" list till in a loop till the end point

Here for adjacency list the first function will be traversing the in edges which will take O(E) time and in the case of ~~matrix~~ matrix to it will traverse for each vertices and edges and it will take time for O(V)

Now for the other function in case of adjacency list it will traverse through all vertices and it will take O(V) and for matrix it will also take ~~to~~ the same time.

So, for adjacency list time complexity is O(V+E) and for matrix it is $O(V^2)$.

For both the tasks in case of adjacency list and for matrix the time complexity is same.

For bfs we see we have go through 9 nodes but for dfs we had to go through only 7 nodes. For bfs we faced ~~much~~ more nodes than dfs because it bfs is a level wise traversal whae dfs is a depth wise traversal. And its not a constant result because it depends on the level and depth of the graph. As hae the endpoint definition is a child node and as he bfs is a parent-priority based and as dfs is a child & priority based algorithm so dfs traveled through less nodes than the bfs.

So Gary will get to the victory road first because of the dfs.