Name : Mollah Md. Saif
ID : 20101416

CSE 221 Lab 03
Section : 10

Answer to the Task no. 4

BFS (Task2):

```
BFS(visited, graph, node, endpoint):
    visited[int(node)-1] = 1                                   ---- 1
    queue.append(node)                                         ---- 1
    while queue not empty:                                     ---- |V|
        m = queue.pop()                                        ---- 1
        Print m                                                ---- 1
        If m == endpoint: break                                ---- 1
        For each neighbour of m in graph:                      ---- deg(m)
            If visited[int(neighbour)-1] == 0:                 ---- 1
                visited[int(neighbour)-1] = 1                  ---- 1
                queue.append(neighbour)                        ---- 1
```

For adjacency ~~matrix~~, List,

while loop have $O(|V|)$

~~while and for loop have $O(\sum deg(v)) = O(2|E|)$~~

Traversing while loop will also traverse every edges of evertices. $\because \sum deg(v) = 2|E| \quad \therefore O(|V| + 2|E|)$

$= O(|V|+|E|)$

For adjacency matrix,

while loop have $O(|V|)$

for loop will traverse every vertices. $\therefore O(|V|)$

$\therefore$ Complexity of BFS, $O(|V|^2)$

DFS (Task 3):

```
DFS_VISIT(vertex, endpoint, visited, printed):
    if endpoint in printed : return ———— 1
    visited[int(vertex)-1] = 1 ———— 1
    printed.append(vertex) ———— 1
    For each neighbour of vertex in graph. ———— deg(neighbour)
        if neighbour in visited: ———— 1
            DFS_VISIT(neighbour, endpoint, visited, printed) —
```

DFS(endpoint, visited, printed):

~~If For each neighbour of~~

For each vertex in graph: ——— |V|

if endpoint in printed: ——— 1

break ——— 1

if vertex not in visited: ——— 1

DFS_VISIT(vertex, endpoint, visited, printed) —

Print all element of printed. ——— 1

For adjacency List,

~~while~~ For loop ~~runs~~ have $O(|V|)$

For every unvisited vertex we run DFS_VISIT.

The time complexity of For loop of DFS_VISIT is

$$O(\Sigma \deg(v \in V))$$

$$= O(2|E|)$$

$$= O(|E|)$$

∴ Total time complexity = $O(|V| + |E|)$

For adjacency matrix,

For loop have time complexity of $O(|V|)$

But for DFS_VISIT it runs for all verteces.

$\therefore$ Total Complexity = $O(|V|^2)$

The DFS algorithm goes to victory road first. Because victory road is extremely far from starting vertex. BFS needs to traverse through all nearest vertices to reach victory road. But with DFS, it goes first to depepest point where it can easily find the victory road.