

## Task - 4

### BFS

visited = []  $\rightarrow O(1)$

queue = []  $\rightarrow O(1)$

bfs(visited, graph, node, endpoint)

visited.append(node)

queue.append(node)

$\rightarrow O(1) \times 3$

while queue not empty  $\rightarrow O(n)$

m = queue.pop()

print m

if m == endpoint

break

$\rightarrow O(1) \times 4$

for each neighbour of m in graph  $\rightarrow O(m)$

if neighbour not in visited

visited.append(neighbour)

queue.append(neighbour)

$\rightarrow O(1) \times 3$

In BFS, when entire tree is traversed then time complexity is  $O(V)$ .

$V$  = number of nodes

$E$  = number of edges

The total of the sizes of the adjacency lists of all nodes in a directed graph is  $E$ . And in case of undirected graph number of edge is twice. Here the given graph is undirected.

The time complexity of BFS will be,

$$O(V) + O(2E)$$

$$= O(V + E)$$

In case of adjacency matrix, BFS time complexity is  $O(V \times V) = O(V^2)$

## DFS

visited = []

dfs-visited (graph, node) —  $O(1) \times 2$

for each node in graph [node] —  $O(n)$

if node not visited

append (i) in visited

dfs-visit (graph, i)

—  $O(1) \times 3$

dfs (graph, endpoint) —  $O(1)$

for each node in graph —  $O(n)$

if node not visited

append (i) in visited

dfs-visited (graph, i)

—  $O(1) \times 3$

let,  $V$  = number of nodes

$E$  = number of edges

Here the given graph is undirected. In case of this, each edge appears twice. And as the tree is traversed its time complexity is  $O(V)$ .

The time complexity of DFS:-

$$O(V) + O(2E) \\ = O(V+E)$$

In case of matrix, time complexity is

$$O(V \times V) = O(V^2)$$

Rival will get to the victory road first because he implemented DFS algorithm to reach victory roads.

Because, in BFS when we see the output which is Ash visited how many countries to reach victory road,

1-2-3-4-5-7-11-6-12

we see Ash visited 7 cities to reach victory road. And in DFS, rival

visited, 5 cities. Crany

1-2-3-4-7-11-12



So we can say that using BFS it took more time to Ash for reaching victory road but rival army took less time as he implemented DFS.