Quiz 3

Design and Analysis of Algorithms

FA19-BCS-037

Mohammad Haris Zia

**COMSATS INSTITUTE OF INFORMATION TECHNOLOGY,**

**ISLAMABAD.**

Department of computer science

Quiz 3

**Instructor:** *Ms. Memoona Malik*

**C. Code & Title:** CSC301Design and analysis of algorithms **Dated:** 1 April 2021

**Maximum Marks**: 20 **Time Allowed:** 30 mins

**Mapped to CLO4**

**Q1: a) Suppose you are provided with a graph of a social networking application. What method would you use to store this graph in computer and why? [3]**

I will use an **undirected graph** in this case. A line between two people will represent that they know each other and a relationship goes both ways. For instance, if a person A knows a person B, that means that person B knows person A as well. If there is no line i.e., an **edge** present between two people that simply signifies that they don’t know each other. The people names represent the **vertices** of the graph and the edges represent the relationship between them. An undirected graph is suitable in this case representing two-way relationships. Also, the vertices connected by edge are referred as **neighbours** or **adjacent** and the number of edges mapped on a certain vertex represent the **degree** of the vertex.

**E.g.:** If a person A does not know a person D i.e., there is no direct edge between the two. We suppose that the person A knows person B who in turn know person C who has a direct edge to person D. That’s how person A can approach person D by traversing through a set of vertices. And since it is the most direct way to approach person D, we refer it as **shortest path**.

Moreover, there also might be cycles in this graph, for instance, if a person A knows person B who in turn knows person C and surprisingly person C knows person A. So, we say that we have a **cycle** in your graph.

1. **What are the advantages and drawbacks of adjacency matrix [5]?**

**Advantages:**

* Adjacency matrices are helpful when we need to quickly check if two nodes have a direct edge or not.
* Time Complexity to List all vertices adjacent to some vertex.
* It can Store weights instead of bits for weighted graphs
* Addition of a new Vertex
* Deletion of a Vertex

**Disadvantages:**

* The main disadvantage is its large memory complexity which is O (n).

1. **Explain the time complexity of depth first search? [4]**

Depth First Algorithm (DFS) traverses a graph in depth and maintains a record of the next visited node via stack.

**Time Complexity:**

***V are the vertices and E represents the edges of a graph.***

In DFS, for each node discover all its neighbors by traversing the adjacency list.

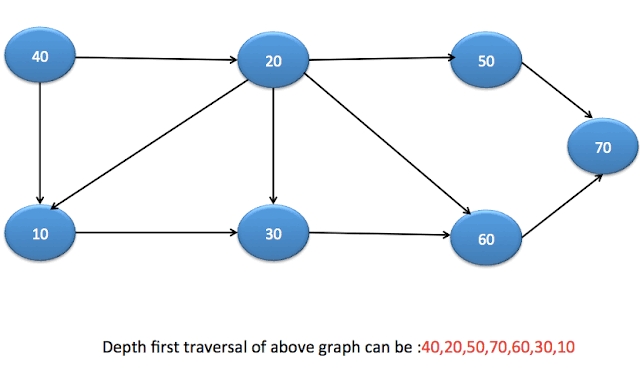
Sum of sizes of adjacency list = Edges (E)

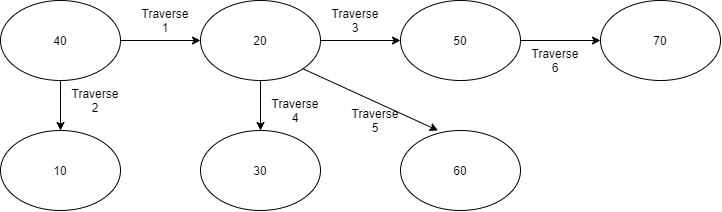
Thus, **time complexity is: O (V) + O(E)**

Where,

**O(V) represents** the time complexity for traversing through the vertices.

**O(E) represents** the DFS visit for neighboring nodes of a vertex v that belongs to V (Vertices)

1. **Consider the following graph and apply breadth first search to traverse all nodes. [8]**



**Breadth First Traversal of the Graph is: 40,20,10,50,30,60,70**

Traverse 1: 40-20

Traverse 2: 40-10

Traverse 3: 20-50

Traverse 4: 20-30

Traverse 5: 20-60

Traverse 6: 50-70'