

EL9343 Homework 08

Spring 2022

Name: Sagar Patel

NETID: SP5894

Question 01

What is the running time of DFS if the graph is given as an adjacency list and adjacency matrix? Justify your running time.

Solution

If the number of the nodes in graph is V and the number of edges in the graph are E , then the running time would be –

Adjacency list: $O(V + E)$

Adjacency matrix: $O(V^2)$

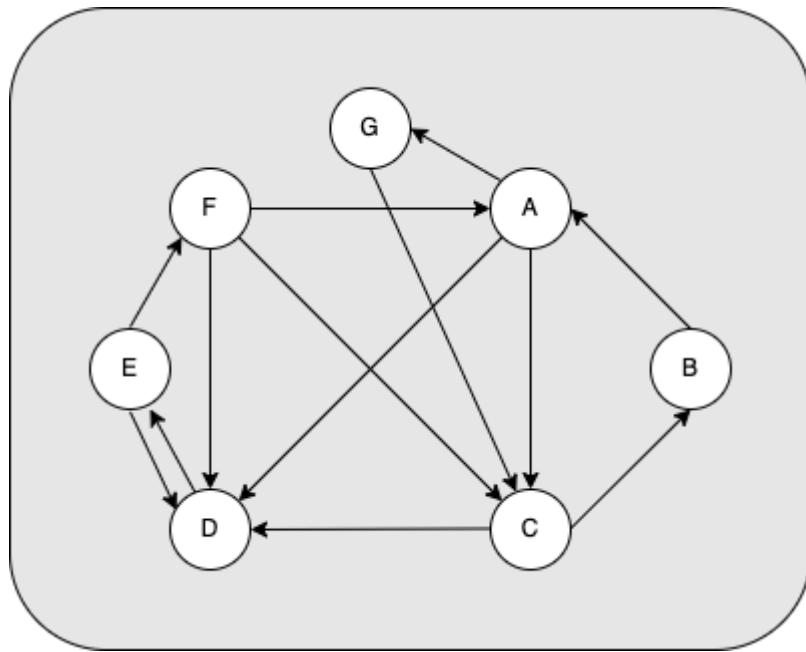
In DFS, we will visit every node once and visit those nodes that are adjacent to them. In both the cases, we first need $O(V)$ time to visit all nodes.

If we use adjacency list, we only need to visit every edge in each list which is $O(E)$. So, the total running time will $O(V) + O(E) = O(V + E)$.

If we use the adjacency matrix, although there are only E edges, once we visit a node (i.e., the i^{th} node), we will visit the element in the i^{th} row of the adjacency matrix. In total, we will visit every element in the adjacency matrix, which is V^2 . So, the total running time will be $O(V^2) + O(V) = O(V^2)$.

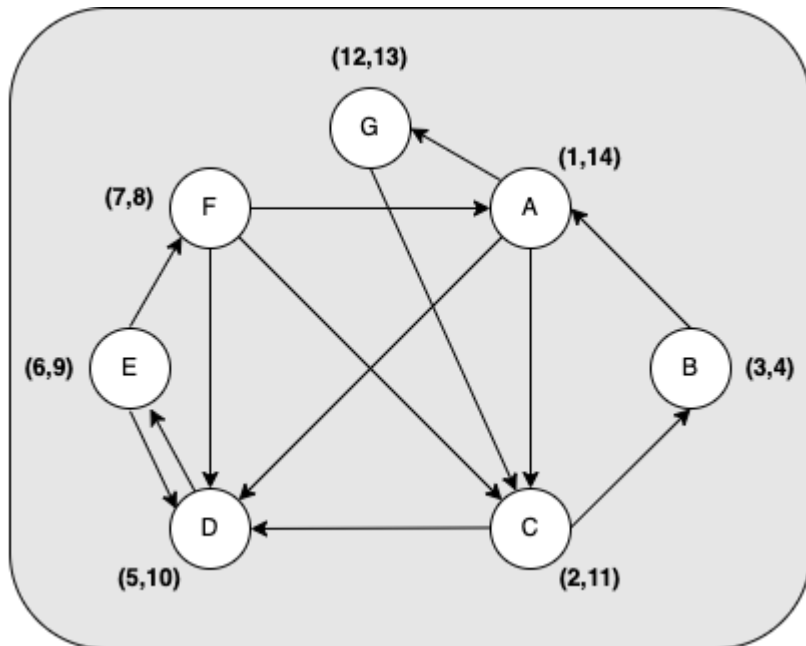
Question 02

Run DFS on the graph below, assume that DFS considers vertices in alphabetical order and the adjacency lists are also alphabetical order. Show the discover times and finishing times of each vertex in the graph.



Solution

The numbers are entered in the format – **(Discover Times, Finishing Times)**.



Question 03

Write a method that takes any two nodes u and v in a tree T , and quickly determines if the node u in the tree is a *descendant* or *ancestor* of node v .

Solution

We could run DFS on a tree T and set the start node as the root and record the discover time and finish time over node. Because T is a tree, the number of edges will be $n - 1$. So, the total running time is $O(V + E) = O(n + n - 1) = O(n)$. As we have proved in the textbook, given any 2

nodes u, v in the DFS tree, if $u.d < v.d < v.f < u.f$, u is the ancestor of v . Else if $v.d < u.d < u.f < v.f$, u is the descendant of v . Because we only need $O(n)$ time to complete DFS and store the discover time and finish time. Other operations are in constant time $O(1)$, so the total running time is $O(n)$. More specifically, we need $O(n)$ time to initialize and $O(1)$ time to determine the relationship between two nodes.

Question 04

Draw the parenthesis structure of the dfs of Figure 1 (start from u, assume that DFS considers vertices in alphabetical order) and see the example parenthesis structure as Figure 2.

Solution

$(u(v(y(xx)y)v)u)(w(zz)w)$