CS21003: Algorithms-I (Theory)
Tutorial – 7 (Graph Traversals)
Date: 26-March-2020

1. Consider a Directed Acyclic Graph (DAG), $G = (V, E)$, where V and E are the set of vertices and edges respectively. Propose suitable algorithms to do the following:

   (i) Extend the topological ordering algorithm to find all possible topological orderings of G. What is the time-complexity?

   (ii) Modify the topological ordering algorithm to find the levels of all the vertices in G. What is the time-complexity?

2. Suppose that the DFS algorithm is applied to a directed graph, $G = (V, E)$, where V and E are the set of vertices and edges respectively. Give an *iterative* DFS algorithm that classifies the edges of G as either tree edges, back edges, forward edges or cross edges and return the count for each of these. [Hint: Use a stack explicitly for writing the iterative DFS.]

3. Given a Directed Acyclic Graph (DAG), $G = (V, E)$, where V and E are the set of vertices and edges respectively and a source vertex $s \in V$ in it, give an algorithm to find the longest distances from s to all other vertices in G. Further, modify the algorithm so that it works for weighted DAG. What are your time-complexity for both the cases?

4. Given two unmarked jugs having capacities C_1 and C_2 liters respectively and a target volume T liters, give an algorithm to find the moves that get exactly T liters in any of the two jugs. Assume that, T is a multiple of $\texttt{GCD}(\texttt{C}_1, \texttt{C}_2)$. [Hint: Use/Modify any of BFS or DFS to solve it.]

5. A Bipartite Graph is a graph whose vertices can be divided into two independent sets, $\mathcal{U}$ and $\mathcal{V}$ such that every edge (u, v) connects a vertex from $\mathcal{U}$ to another vertex $\mathcal{V}$ or vice-versa. In other words, for every edge (u, v), either $u \in \mathcal{U}$ and $v \in \mathcal{V}$, or $u \in \mathcal{V}$ and $v \in \mathcal{U}$. We can also say that there is no edge that connects vertices of same set. Now, given a graph, $G = (V, E)$, where V and E are the set of vertices and edges respectively, propose an algorithm to find that G is a bipartite graph or not. [Hint: Use/Modify any of BFS or DFS to solve it.]