# ESO207: Data Structures and Algorithms (Practice Problems – II)

**Question 1: Cycle Detection in a graph**

(a) Using BFS design an O(n) time algorithm to determine if a given undirected graph has a cycle.

(b) Using DFS design an O(m + n) time algorithm to determine if a directed graph has a cycle.

**Question 2: On size of connected component**

You are given a graph $G = (V, E)$ and two vertices $u$ and $v$. Furthermore, you are told that $u$ and $v$ belong to different connected components. The size of a connected component is defined as the number of edges present in that component. You have to determine whether the component of $u$ is smaller than the component of $v$. Design a simple algorithm for this problem whose running time is of the order of the size of the smaller component.

**Question 3: Topological ordering in DAG**

Given a directed acyclic graph of $n$ vertices and $m$ edges, design an $O(n+m)$ time algorithm to compute its topological ordering (see textbook for the definition of topological sorting).

**Question 4: Structure for answering ancestor descendent query**

Given a rooted tree on $n$ vertices where the vertices are numbered from 0 to $n-1$. Let $r$ be the index of the root node. Design an $O(n)$ size data structure using which it takes $O(1)$ time to answer the following query for any $0 \leq i, j < n$.

**IsAncestor(i, j)**: Is $i$ an ancestor of $j$?

**Question 5: Determine if a tree is DFS tree**

We know that there may be many possible DFS trees for a graph depending upon the start vertex and the order in which we explore the neighbors of each vertex.

Given a connected graph $G = (V, E)$, you are given a rooted tree $T$ such that each edge of this tree is present in $E$. Design an efficient algorithm to determine if $T$ is a DFS tree of $G$.

**Question 6: Constructing a DAG**

You are given an undirected graph $G = (V, E)$ which is connected. $G$ may have many cycles in it. You want to assign a direction to each edge in the graph so that the resulting directed graph does not have a cycle. Design an efficient algorithm for this task.

**Question 7: Euler tour of a graph**

**(Difficult)** A connected graph $G$ is said to be a Eulerian graph if starting from any vertex $v$, it is possible to make a walk on the graph which terminates at $v$ such that each edge is visited exactly once (though a vertex may be visited multiple times). The corresponding tour is called an Euler tour of G. Design an $O(m + n)$ time algorithm to output an Euler tour of a graph $G$, if it exists.

(Hint: You may try using DFS along with the following characterization of an Eulerian graph. A connected graph is Eulerian if and only if the number of edges incident on each vertex is even.)

**Question 8: Computing $k$-smallest elements**

Given an array $A$ storing $n$ numbers. Design an $O(n + klogn)$ time algorithm to compute $k$-smallest elements from $A$. You may modify the array $A$ if you wish.

**Question 9: Comparison with $k$-th smallest element**

You are given a binary heap $H$ of size $n$, a number $x$, and a positive integer $k$. Design an $O(k)$ time algorithm to determine if $x$ is smaller than $k$-th smallest element in $H$. You may use $O(k)$ extra space.

**Question 10: Equivalent definition of tree**

Prove that the following are equivalent for a graph $G = (V, E)$ where $|V| = n$ and $|E| = m$.

1. $G$ is connected and acyclic.
2. $G$ is connected and $m = n - 1$.
3. There is a unique path between every pair of vertices in $G$.