Graph Theory Assignment - 1
DISCLAIMER: PLEASE DO NOT ATTEND THE NEXT COMPETITIVE PROGRAMMING CLASS WITHOUT COMPLETING THIS ASSIGNMENT AS YOU WONT UNDERSTAND ANYTHING.

Reading material:
1. How to represent graphs (pre-requisites : vectors in C++). Do quest at the last of this tutorial.
2. Depth First Search (DFS) (pre-requisites: recursions, stacks in C++). Do quest at the last of this tutorial.
3. Breadth First Search (BFS) (pre-requisites: queue in C++). Do quest at the last of this Tutorial.

Write your solutions in C++ or Java or .txt (for non-coding quests) and compress all files in a folder and then mail the folder to cyberlabs.ism@gmail.com by 11:59 PM, 21st Oct. The format of each file should be question_number.cpp or question_number.java or 1.txt (for quests whose answers are not codes). (E.g. for quest 1 => 1.txt; for quest 2 => 2.cpp or 2.java). The name of the file should be firstName_admnNumberInCapitals.zip.
The code must be ready to compile and run. You can assume that cases like graph is empty will not exist.
NOTE: The following assignment will be a brain opener for those who have not started graph theory yet. So, please be honest to yourselves and try your best. All the best. The assignment is compulsory for everyone who will be attending competitive classes post DP.
Ideally, this should take 2 days of time for a beginner and 2-6 hours for someone who knows DFS and BFS already, depending upon the level of the person.
Index: 1. V: No of vertices of graph
　　　　2. E: No of edges of graph
Questions:
1. What is the difference between a graph, a tree and a DAG.
　　For quests 2 to 8, consider suitable V, E and each edge (a,b).
2. Implement BFS for a tree taking 1 as root.
3. Implement DFS for a Directed graph taking 1 as root.
4. Implement DFS for an Undirected graph taking 1 as root.
5. Check if an undirected graph is a tree or not.
6. Check if an undirected graph is bipartite or not. (Read definition of bipartite graph yourself, but do not read the actual solution)
7. Check if an undirected graph has a cycle
8. Check if a directed graph has a cycle
9. Explain the time complexity of BFS for a graph
10. Explain the time complexity of DFS for a graph
11. Define Euler tour, Hamiltonian cycle, topological ordering in your own words, with thinking real life examples on ur own(also mention them in answer). Craft an example yourself for understanding and mention it.
12. Can multiple topological orderings exists? Take examples of DAGs and then try to answer.