ESO207A: Data Structures and Algorithms	
Homework *3: Basic Graph Algorithms*	HW Due Date: Oct 15, 2018

Instructions.

1. Solutions have to be submitted only on Gradescope and within the deadline. For this, each problem should start on a new sheet. If this is not followed then it would not be possible to grade your answers.
2. For each problem, write your name, Roll No., problem number, the date and the names of any students with whom you collaborated. Remember that you must write the answer and the algorithm in your own words.
3. For questions asking for graph algorithms, give (a) a clear description of the algorithm in English and/or pseudo-code, where, helpful, (b) a proof/argument of the correctness of the algorithm, and, (c) an analysis of the running time of the algorithm.
4. You can refer to the standard algorithms covered in the class and use them as appropriate sub routines.
5. For a graph $G = (V, E)$, let $n = |V|$ and $m = |E|$.

   Full marks will be given only to correct solutions which are described clearly. Convoluted and unclear descriptions will receive *low marks*.

Problem 1. Given a directed graph $G = (V, E)$ defined the graph $G^R = (V, E^R)$ with all edge directions reversed, that is, $E^R = \{(v, u) | (u, v) \in E\}$.

1. Suppose G is represented as an adjacency matrix. Give an $O(|V|^2)$ algorithm for finding the adjacency matrix representation of G^R. (4)
2. Suppose G is represented as an adjacency list. Given an $O(|V| + |E|)$ algorithm for finding the adjacency matrix representation of G^R. (4)
3. A source vertex in G is a vertex s such that there is no vertex $v \in G$ such that $(v, s) \in E$. A sink vertex in G is a vertex u such that there is no vertex $v \in G$ such that $(u, v) \in G$. Relate the source and sink vertices of G and G^R. Relate the strong components of G with the strong components of G^R. (2+2)

Problem 2. A connected graph $G = (V, E)$ is said to be bi-partite if V can be partitioned into two non-empty and disjoint partitions $V = V_1 \cup V_2$ such that any edge $\{u, v\}$ in E has $u \in V_1$ and $v \in V_2$. That is, there are no edges among vertices in V_1 or among vertices in V_2. Consider the following test for bi-partition. Run BFS on G starting from vertex s. Let $u.d$ be the shortest distance (as per BFS) from s to u.

1. Show that G is bipartite iff for every edge $\{u, v\} \in E$, $|u.d - v.d| = 1$. (12)

2. Extend the BFS algorithm slightly to give an $O(|V|+|E|)$ algorithm to test if G is bi-partite. (12)

Problem 3. Give an $O(|V|+|E|)$ algorithm that takes a directed graph and a given edge $e = (u, v)$ and tests if there is a cycle involving e in the graph. (20)

Problem 4. Given a graph $G = (V, E)$, consider the following alternative outline for finding a topological order that repeatedly removes source nodes from G.

Repeat until the graph is empty
    Find a source, output it and delete it.

Design an $O(|V| + |E|)$ time algorithm to implement this (alternative) outline and prove your complexity. (22)

Problem 5. Given a directed graph $G = (V, E)$, design an algorithm to find a vertex u that has paths to all vertices in V in G. (22)