Homework 5
Com S 311
Due: Nov 4, 11:59PM

Late Submission Due: Nov 5, 11:59PM (25% penalty)

Outcomes.

- Understand BFS/DFS

Please see guidelines for submission instructions.

This is a short homework with 3 problems. Each problem is worth 50 points. Unless stated otherwise the input representation of graphs is *adjacency list*.

1. Let $G = (V, E)$ be a directed graph. Define a graph $G^2 = (V', E')$ as follows: $V' = V$; $\langle u, v\rangle \in E'$ if there is a a path of length 2 between u and v in G.

   - Suppose that a directed graph G is given as adjacency list. Give a $O(mn)$-time algorithm to compute G^2. Derive the time bound. Here m is number of edges in G and n is number of vertices in G.
   - Suppose that a directed graph G is given as adjacency matrix. Given an algorithm to compute G^2. Derive the run time of your algorithm.

2. Let $G = (V, E)$ be a directed graph where $V = \{1, 2, \cdots, n\}$ such that n is odd; I.e n $= 2k+1$ for some $k > 0$. Given a vertex v, let TO_v be the set of all vertices from which there is path to v. Let $FROM_v$ be the set all vertices for which there is a path from v. I.e,

   $$TO_v = \{u \mid \text{ There is a path from } u \text{ to } v\},$$

   $$FROM_v = \{w \mid \text{There is a path from } v \text{ to } w\}.$$

   A vertex v is *center vertex* of G if all of the following conditions hold:

   - $|TO_v| = |FROM_v| = k$. I.e, both TO_v and $FROM_v$ have exactly k vertices.
   - $TO_v \cap FROM_v = \emptyset$. I.e, TO_v and $FROM_v$ are disjoint.

   Give an algorithm that gets a graph G (with odd number of vertices) as input and determines if the graph has a center vertex or not. If the graph has a center vertex, then the algorithm must output it. Describe your algorithm, prove the correctness, and derive the time bound. Your grade partly depends on the efficiency of your algorithm.

3. Let $G = (V, E)$ be an undirected, connected graph. A vertex $v \in V$ is *bridge vertex* if removal of v (and edges incident on v) makes the graph disconnected.

    - Suppose that $v \in V$ be a bridge vertex. Is there a vertex $u \in V$ such that if we perform DFS on G starting at u, the vertex v will be a leaf node in the resulting DFS tree?
    - Given an $O(m + n)$-time algorithm that gets a graph G (which is undirected and connected) as input and outputs a vertex v that is not a bridge vertex. Describe your algorithm, derive the time bound, prove the correctness of your algorithm.

GUIDE LINES:

- It is important to know whether your really know! For each problem, if you write the statement "I do not know how to solve this problem" (and nothing else), you will receive 20% credit for that problem. If you do write a solution, then your grade could be anywhere between 0% to 100%. To receive this 20% credit, you must explicitly state that you do not know how to solve the problem.

- You must work on the homework problems on your own. You should write the final solutions alone, without consulting any one. Your writing should demonstrate that you understand the proofs completely.

- When proofs are required, you should make them both clear and rigorous. Do not hand waive.

- If you hand writing is not legible, then your homework will not be graded.

- Any concerns about grading should be made within one week of returning the homework.

- **Please submit your HW via blackboard. If you type your solutions, then please submit pdf version. If you hand-write your solutions, then please scan your solutions and submit a pdf version. Please make sure that the quality of the scan is good, and your hand writing is legible. Name your file must be $YourNetID$-$HW5.pdf$. For example, if your net id is `bondj`, then the file name should be *bondj-HW5.pdf*. HW's submitted in incorrect format (non pdf, incorrect file name etc) will incur a penalty of 30%**