CMPSC 465 Data Structures & Algorithms
Spring 2021 Paul Medvedev and Chunhao Wang

HW 5

Due February 25, 10:00 pm

Instructions: You may work in groups of up to three people to solve the homework. You must write your own solutions and explicitly acknowledge up everyone whom you have worked with or who has given you any significant ideas about the HW solutions. You may also use books or online resources to help solve homework problems. All consulted references must be acknowledged.

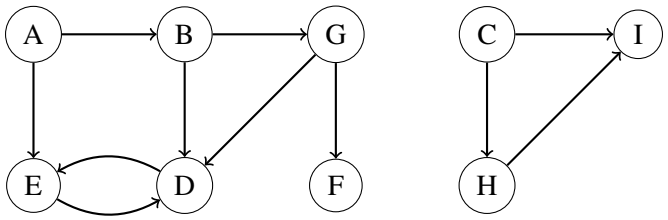
You are encouraged to solve the problem sets on your own using only the textbook and lecture notes as a reference. This will give you the best chance of doing well on the exams. Relying too much on the help of group members or on online resources will hinder your performance on the exams.

Late HWs will be accepted until 11:59pm with a 20% penalty. HWs not submitted by 11:59pm will receive 0. There will be no exceptions to this policy, as we post the solutions soon after the deadline. However, you will be able to drop the three lowest HW grades.

For the full policy on HW assignments, please consult the syllabus.

1. **(0 pts.) Acknowledgements.** The assignment will receive a 0 if this question is not answered.
   (a) If you worked in a group, list the members of the group. Otherwise, write "I did not work in a group."
   (b) If you received significant ideas about the HW solutions from anyone not in your group, list their names here. Otherwise, write "I did not consult without anyone my group members".
   (c) List any resources besides the course material that you consulted in order to solve the material. If you did not consult anything, write "I did not consult any non-class materials."

2. **(12 pts.) DFS basics**



   Run DFS starting from node A, trying to visit nodes alphabetically (e.g. given a choice between nodes D and F, visit D first).
   (a) List the nodes in the order you visit them (so each node should appear in the ordering exactly once).
   (b) List each node with its pre- and post-number. The numbering starts from 1 and ends at 18.
   (c) Label each edge as **T**ree, **B**ack, **F**orward or **C**ross.

3. **(12 pts.) Pre and Post Processing**

(a) Either prove or give a counterexample: if $\{u, v\}$ is an edge in an undirected graph, and during depth-first search $post(u) < post(v)$, then v is an ancestor of u in the DFS tree.

(b) You are given a tree $T = (V, E)$ (in adjacency list format), along with a designated root node $r \in V$. Recall that u is said to be an *ancestor* of v in the rooted tree if the path from r to v in T passes through u. You wish to preprocess the tree so that queries of the form "is u an ancestor of v?" can be answered in constant time. The preprocessing itself should take linear time. How can this be done?

4. (10 pts.) Application of DFS

We are given a directed graph $G = (V, E)$, where $V = \{1, \ldots, n\}$, i.e. the vertices are integers in the range 1 to n. For every vertex i we would like to compute the value $m(i)$ defined as follows: $m(i)$ is the smallest j such that vertex j is reachable from vertex i. (As a convention, we assume that i is reachable from i.) Show that the values $m(1), \ldots, m(n)$ can be computed in $O(|V| + |E|)$ time.

(Hint1: Make use of the reverse graph G^R)

(Hint2: You should probably use DFS as a subroutine. Feel free to slightly modify DFS if it helps.)