CMPSC 465 Data Structures & Algorithms
Spring 2021 Paul Medvedev and Chunhao Wang **HW 4**

Due February 18, 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. **(16 pts.) Divide-and-Conquer** An array $A[1\dots n]$ is said to have a *majority element* if more than half of its entries are the same. Given an array, the task is to design an efficient algorithm to tell whether the array has a majority element, and, if so, to find that element. The elements of the array are not necessarily from some ordered domain like the integers, and so there can be no comparisons of the form "is $A[i] > A[j]$?" (Think of the array elements as GIF files, say.) However you can answer questions of the form: "is $A[i] = A[j]$?" in constant time. Design a dynamic programming algorithm to solve this problem in $O(n \log n)$ time. You may assume that n is a power of two. (Hint: the divide step is simple: just split the array into the left half and right half. How does knowing the majority elements (if they exist) of the two halves help you figure out the majority element in the whole array? Consider all the possibilities.)
   (a) Describe the algorithm, either in pseudo-code or English.
   (b) Argue why your algorithm is correct
   (c) Do a running time analysis of your algorithm.

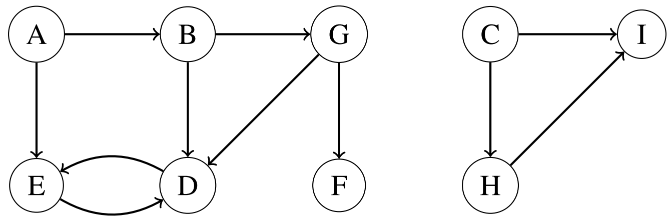
3. **(12pt pts.) Reverse graph:** Let $G = (V,E)$ be an directed graph. The reverse of G is another directed graph, $G^R = (V,E^R)$. In G^R, the vertices are the same as in G, but the edges are reversed. In other words, $E^R = \{(v,u) : (u,v) \in E\}$.

(a) If G is stored in adjacency list format, give an $O(|V|+|E|)$ time algorithm to create an adjacency list representation of G^R. Argue why the running time is $O(|V|+|E|)$.

(b) If G is stored in adjacency matrix format, given a $O(|V|^2)$ time algorithm to create an adjacency matrix representation of G^R. Argue why the running time is $O(|V|^2)$.

4. (20 pts.) Graph Basics:

Give a short answer to each of the questions below.



(a) Write down the adjacency list representation of the graph above.

(b) What is the most number of edges that an undirected graph can have? Give an exact answer in terms of $|V|$, without asymptotic notation. You may assume that there are no parallel edges, i.e. for any two vertices u and v, there is at most one edge $\{u,v\}$.

(c) For each vertex v_i in an undirected graph, let d_i be its *degree*, which is the number of edges incident to it. Show that $\sum_{i=1}^{|V|} d_i$ must be even. You may assume that there are no self-loops, i.e. edges of the form (u,u).

(d) Suppose your boss is trying to solve a problem on a graph, and you present them with two algorithms. The first runs in time $\Theta(|V||E|)$. The second runs in time $\Theta(|V|^2 \log|V|)$. Your boss is happy you know the running times, but they seem confused as to which one is better. So they ask you to tell them, in plain English, which algorithm will run faster. What do you answer?

(e) Suppose instead that the first algorithm runs in time $\Theta(|E|\log|V|)$ and the second runs in $\Theta(|E|\log|E|)$. What do you tell your boss then?