Homework #5
Algorithms I
600.463
Spring 2017

Due on: Saturday, March 18th, 11:59pm
Late submissions: will NOT be accepted
Format: Please start each problem on a new page.
Where to submit: On Gradescope, under HW5
Please type your answers; handwritten assignments will not be accepted.
To get full credit, your answers must be explained clearly,
with enough details and rigorous proofs.

March 8, 2017

Problem 1 (20 point)

Let $G = G(V, E)$ be a directed graph represented by an adjacency list. G is a bipartite graph if it is possible to partition the vertices of G into two disjoint sets, i.e. $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$ such that there are no edges between vertices in the V_1 and there are no edges between vertices in the V_2. Design an efficient algorithm that works in $O(|E| + |V|)$ time and checks if G is bipartite. Prove the correctness of your algorithm and analyze the running time.

Problem 2 (20 points)

Problem 2.1 (10 points)

Suppose we wish not only to increment a counter but also to reset it to zero (i.e., make all bits in it 0). Counting the time to examine or modify a bit as $\Theta(1)$, show how to implement a counter as an array of bits so that any sequence of n $INCREMENT$ and $RESET$ operations takes time $O(n)$ on an initially zero counter. (Hint: Keep a pointer to the high-order 1.)

Problem 2.2 (10 points)

Design a data structure to support the following two operations for a dynamic multiset S of integers, which allows duplicate values:

$INSERT(S, x)$ inserts x into S.

$DELETE\text{-}LARGER\text{-}HALF(S)$ deletes the largest $\lceil |S|/2 \rceil$ elements from S.

Explain how to implement this data structure so that any sequence of m $INSERT$ and $DELETE\text{-}LARGER\text{-}HALF$ operations runs in $O(m)$ time. Your implementation should also include a way to output the elements of S in $O(|S|)$ time.