Problem Set 1

Data Structures and Algorithms, Fall 2019

Due: September 12, in class.

From CLRS

Exercise 2.1-3, 2.2-2, 2.2-4. Problem 2-3. Exercise 3.1-1, 3.1-8, 3.2-4, 3.2-5. Problem 3-2, 3-3.

One Bonus Problem

*This is a **bonus** problem, which means you are NOT required to solve it. However, this is a very interesting problem, and part (a) of it is a common interview question. Part (b), on the other hand, reveals why the "trick" you employed in part (a) works, and how to generalize it to solve similar problems.*

(a) You are given a multiset S containing $n = 2k+1$ elements. Each element is a 32-bit positive integer. Among these elements, there are k elements each appearing twice, and one element that appears only once. Devise an algorithm which can find the element that appears once. Your algorithm should use $O(n)$ time and $O(1)$ space. (*Hint: consider bit operations.*)

(b) You are given a multiset S containing $n = 3k+1$ elements. Each element is a 32-bit positive integer. Among these elements, there are k elements each appearing three times, and one element that appears only once. Devise an algorithm which can find the element that appears once. Your algorithm should use $O(n)$ time and $O(1)$ space. (*Hint: interpret the bit operation you used for solving part (a) as modulo operation.*)