Assignment 5  
Computer Science 231  
Spring 2017  
Due: Tuesday, March 7th

Reading. CLRS Chapter 8, pages 191 -- 212

CLRS Exercise 8.1-1. What is the smallest possible depth of a leaf in a decision tree for a comparison sort under the assumption that all elements are distinct?

CLRS Exercise 8.1-4. You are given a sequence of \( n \) elements to sort. The input sequence consists of \( n/k \) subsequences, each containing \( k \) elements. The elements in a given subsequence are all smaller than the elements in the succeeding subsequence and larger than the elements in the preceding subsequence. Thus, all that is needed to sort the whole sequence of length \( n \) is to sort the \( k \) elements in each of the \( n/k \) subsequences.

Show an \( \Omega(n \lg k) \) lower bound on the number of comparisons needed to solve this variant of the sorting problem.  (Hint: It is not rigorous to simply combine the lower bounds for the individual subsequences.)

CLRS Exercise 8.2-4. Describe an algorithm that, given \( n \) integers in the range 0 to \( k \), preprocesses its input and then answers any query about how many of the \( n \) integers fall into a range \([a..b]\) in \( O(1) \) time. Your algorithm should use \( \Theta(n + k) \) preprocessing time.

CLRS Exercise 8.3-4. Show how to sort \( n \) integers in the range 0 to \( n^2 - 1 \) in \( O(n) \) time.

CLRS Exercise 9.1-1. Show that the second smallest of \( n \) elements can be found with \( n + \lceil \lg n \rceil - 2 \) comparisons in the worst case.  (Hint: Build a "tournament tree" over the elements in which each node has a winner (smaller element) and a loser (larger element). In other words, construct a tree whose leaves are the \( n \) elements and each of whose nodes holds the minimum value of the leaves of the subtree rooted at that node.) Be careful that your algorithm accounts for all comparisons performed.