<u>Due</u>: Mar 12th at the start of class

Homework #3

CMSC351 - Spring 2013

PRINT Name____:

- *Grades depend on neatness and clarity.*
- *Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should ALWAYS <u>prove the correctness of your algorithms</u> either directly or by referring to a proof in the book.*
- *<u>Write your answers in the spaces provided.</u> If needed, attach other pages.*
- *The grades would be out of 16. Four problems would be selected, and everyone's grade would be based only on those problems. You will also get 4 bonus points for trying to solve all problems.*

1. [Prob 6.4,Pg 176] Construct an example for which interpolation search will use $\Omega(n)$ comparisons for searching in a table of size *n.*

2. Consider the following sequence of numbers as the input:

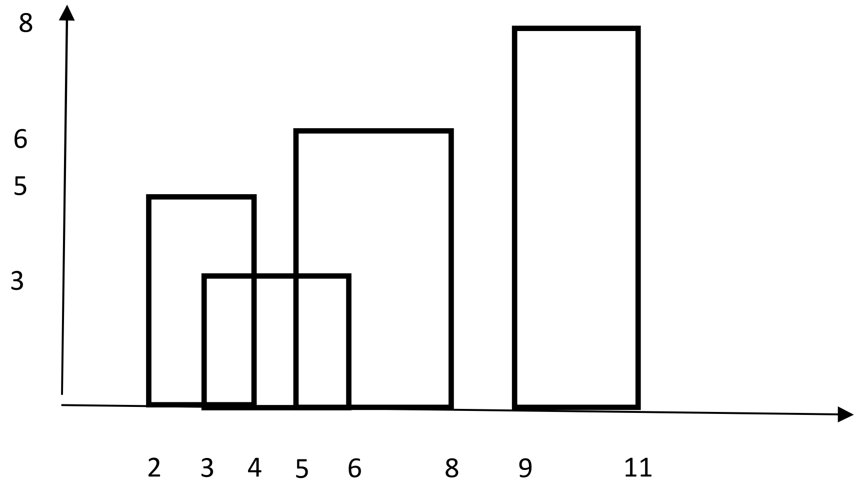
6 2 8 5 10 9 12 1 15 5

a) Sort the sequence using quicksort algorithm (Fig 6.11). Rewrite the sequence after each swap operation. What is the number of comparisons between elements of the input array used by quicksort?

b) Count the number of comparisons between elements of the input array if we use mergesort (Fig 6.7).

3. You are given an array of n elements, and you notice that some of them are duplicates, that is, they appear more than once in the array. Show how to remove all duplicates from the array in time $O(n \log n)$.

4. **The Skyline Problem**: Consider a sequence of n buildings in 2 dimensional space. The i^{th} building is specified by a triple (l_i, h_i, r_i) where l_i, r_i are the starting and ending x-coordinates and h_i is the height. Construct the skyline of these buildings in $O(n \log n)$ time, which is a sequence of values of x-coordinates and heights connecting them arranged by order of the x-coordinates. See the example below. We write the heights in red color.



For this example the Input is (2,5,4), (3,3,6), (5,6,8) and (9,8,11)
The output should be (2,5,4,3,5,6,8,0,9,8,11)

5. [Prob 6.23,Pg 177] Given two sets S_1 and S_2, and a real number x, find whether there exists an element from S_1 and an element from S_2 whose sum is exactly x. The algorithm should run in time $O(n \log n)$, where n is the total number of elements in both sets.

6. [Prob 6.24,Pg 177] Design an efficient algorithm to determine whether two sets with sizes *n and m* are disjoint. State the complexity of your algorithm. Make sure to consider the case where *m* is substantially smaller than *n.*

7. Show how to sort n integers in the range 0 to $n^2 - 1$ in $O(n)$ time.

   (Hint: Consider the numbers in base n, i.e., each number x in the range $[0 \,..\, (n^2 - 1)]$ can be considered in the form $(ab)_n$ where $a = x \ div \ n$ and $b = x \ mod \ n$.)