COMP 2119A Introduction to Data Structures and Algorithms
Assignment Three
Due Date: 25 November 2016 midnight

Assignment box: A3 (also submit a scanned copy to moodle)

[Note that for the questions involving algorithm design, before you present your algorithm, try to describe your idea first!
**More marks will be given to faster algorithms.**]

** NO LATE ASSIGNMENT will be accepted **

1) (a) [10%] Insert the following numbers into an initially empty AVL tree one by one. Show the resulting tree after each insertion. Then, delete 98, 120. Show also the resulting tree after each deletion.

38, 25, 12, 120, 57, 96, 98, 43

(b) [10%] We are given $n$ numbers (not necessarily distinct). We know that there are only $k$ distinct values for the number. Design an algorithm to return a number with the highest frequency. Analyze the time complexity of your algorithm (you got higher score for faster algorithm).

2) Consider the average case analysis for binary search. Let $A[1..n]$ be the given array of sorted distinct numbers and let $n = 2^k$ for some integer $k$. Consider a *successful* search (i.e., the number you want to search exists in the array $A$ and there are only $n$ possible cases for input).

(a) [4%] How many case(s) requiring only <u>one</u> comparison to locate the given number? How many cases requiring exactly two comparisons to locate the given number?
(b) [6%] How many case(s) requiring exactly $i$ comparisons to locate the given number ($1 \le i \le k$).
(c) [10%] Hence, or otherwise, compute the average case complexity for a successful search in binary search assuming that all $n$ cases are equally likely

3) [20%] Sort the following numbers in increasing order using (a) bubble sort; (b) insertion sort; (c) selection sort; (d) quick sort (always pick the last element as the pivot). Show the resulting list of numbers after each round for (a) – (c). For (d), show the resulting list after selecting the pivot and rearranging the numbers based on the pivot.

188, 157, 122, 148, 192, 136, 167, 31, 138, 202, 126

4) (a) [10%] Insert the numbers in Q(3) into an initially empty min-heap one by one. Show the resulting heap after each insertion. Show the resulting heaps after executing the Extract-Min operation twice.

(b) [10%] Given $n$ distinct numbers (not sorted), design an $O(n + k \log n)$ algorithm to retrieve the $k$ smallest numbers.

5) [15%] Consider each of the following cases. If it is correct, give a proof. If it is incorrect, give a counter-example.
(a) In a binary search tree, if node $u$ has two children, its predecessor has no left child.

(b) In a binary search tree, if node $u$ has two children, its predecessor has no right child.

(c) Given the preorder and postorder traversal of a binary tree, we can uniquely determine the topology of the binary tree (i.e, there does NOT exist two different binary trees with the same preorder and postorder traversals).

6) [5%]
- (a) Is the assignment (1) too difficult; (2) too easy; (3) about right?
- (b) How many hours you spend on the assignment?
  - (1) Less than 5 hours
  - (2) 5 – 10 hours
  - (3) 10 – 20 hours
  - (4) More than 20 hours
- (c) [Self assessment] Do you consider yourself understand the topics of this assignment?
  (1) Yes; (2) not 100% sure; (3) No
  If your answer is (2) or (3), please elaborate (at least indicate which part you do not understand).
- (d) Other comments?

--- End of Assignment ---