EC 504 – Fall 2019 – Homework 3

Due Thursday, Oct 22, 2019 in the beginning of class. Coding problems submitted in the directory `/projectnb/alg504/yourname/HW3` on your SCC account by Friday Octo 22, 11:59PM.

Reading Assignment: CLRS Chapters 12, 13, Sec 16.3, B.2 (relations) and Introduction to Graphs: Chapter 22 and B.4

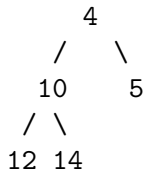
1. (15 puts) Determine whether the following statements are true or false, and explain briefly why.

   (a) A heapsort algorithm for a given list first forms a min-heap with the elements in that list, then extracts the elements of the heap one by one from the top. This algorithm will sort a list of n elements in time of O(log(n)).

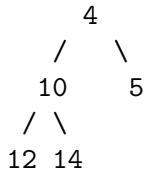
   (b) A connected undirected acyclic graph is a binary tree.

   (c) A binary heap of n elements is a full binary tree for all possible values of n.

   (d) The following tree is a valid min-heap.



   (e) The following tree can appear in a binomial min-heap. .



   (f) Given a array of positive integers $a[i]$, maximizing $\sum i * a[i]$ over permutations of the array requires sorting $a[i]$ in assigning order.

2. (20 Pts) Given the following list of elements,

$$35, 22, 11, 98, 55, 66, 77, 80, 85, 90, 95$$

   (a) Draw the sequence of binary min-heaps which results from inserting the following values in the order in which they appear into an empty heap.

   (b) Compare this answer with inserting them into the BST tree.

   (c) Compare this answer with inserting them into the AVL tree.

3. (20 Pts) You are interested in compression. Given a file with characters, you want to find the binary code which satisfies the prefix property (no conflicts) and which minimizes the number of bits required. As an example, consider an alphabet with 8 symbols, with relative weights (frequency) of appearance in an average text file give below:

alphabet:	A	L	G	O	R	I	T	H
weights:	68	20	5	30	18	15	19	12

(a) Determine the Huffman code by constructing a tree with **minimum external path length**: $\sum_{i=1}^{8} w_i d_i$. (Arrange tree with smaller weights to the left.)

(b) Identity the code for each letter and list the number of bits for each letter and compute the average number of bits per symbole in this code. Is it less than 3? (You can leave the answer as a fraction since getting the decimal value is difficult without a calculator.)

(c) Give an example of weights for these 8 symbols that would saturate 3 bits per letter. What would the Huffman tree look like? Is a Huffman code tree always a full tree?

4. (20 Pts) Given the following list of $N = 10$ elements.

$$28 \quad 14 \quad 7 \quad 4 \quad 6 \quad 30 \quad 36 \quad 33 \quad 10 \quad 40$$

(a) Insert them sequentially into a BST (Binary Search Tree). Compute the total height $T_H(N)$ and the total depth $T_D(N)$, where H is the height of the root. (Note the height of a node is the longest path length to a leaf whereas its depth is its unique distance from the root.)

(b) Find H, $T_H(N)$, $T_D(N)$ and check the sum rule:

$$T_H(N) + T_D(N) \leq HN$$

(c) Insert them sequentially into an empty AVL tree, restoring the AVL property after each insertion. Show the AVL tree which results after each insertion and name the type of rotation (RR or LL zig-zig or versus RL or LR zig-zag).

(d) Has the final AVL tree decreased the total height $T_H(N)$ and the total depth $T_D(N)$? What are the new values? What is the new value of $T_H(N) + T_D(N)$?

5. **Coding Exercise:** (25 Pts) You are give a code that runs a binary search tree (BST). The BST code is has the following function

```
SearchTree MakeEmpty( SearchTree T );
Position Find( ElementType X, SearchTree T );
Position FindMin( SearchTree T );
Position FindMax( SearchTree T );
SearchTree Insert( ElementType X, SearchTree T );
SearchTree Delete( ElementType X, SearchTree T );
ElementType Retrieve( Position P );
```

declared in `tree.h` and defined in `tree.cpp`

As in the case of HW2 HeapSort the main program `mainBST.cpp` has the following function allows you to insert into the binary search tree from either an input file or form generate internally a random sequence. You should use the same input list `List100.txt` and `List100K.txt` used in HW2. Note `n = InputList[0]` is the size of the list NOT an element to be used in the BST which are `InputList[i] for i = 1, ..., n`

The problem is to

```
(1)Read into InputList[]  the files List100.txt first and then  List100K.txt
(2)Build the heap from the first half (i = 1,...n/2)
(3)Delete the even ones (i= 2,4,.., n/2)
(4)Insert the second half i = n/2 +1,..., n
(5) Find the Max and Min in this BST
(6) Implement a function that calculate the height of the BST.
```

Extra Credit starting with random list generated internally.

```
(1) Consider the BST properties  for n = 8, 16, 32, 64, ... 4048 (or larger?)
(2) Can you see and plot the heights average over random each?
(3) For n= 1024  show that random deletion/insertions increases height.
```