EC 504 – Fall 2020 – Homework 4

Due Thursday, Oct 22, 2020, submitted in the directory `/projectnb/alg504/username/HW4` **on your SCC account by 11:59PM.**

Reading Assignment on GitHub:
Sorting_Data_Structure (Chapters 6, 7) and
Trees.pdf (12, 13) and TreeMath_B.pdf (Appendix B .5)

1. (20 pts) Determine whether the following statements are true or false, and explain briefly why.

   (a) If doubling the size ($N \to 2N$) causes the execute time $T(N)$ of an algorithm to increase by a factor of 4, then $T(N) \in O(4N)$.

   (b) The height of a binary tree is the maximum number of edges from the root to any leaf path. The maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$.

   (c) In a binary search tree with no repeated keys, deleting the node with key x, followed by deleting the node with key y, will result in the same search tree as deleting the node with key y, then deleting the node with key x.

   (d) Inserting numbers 1, . . . , n into a binary min-heap in that order will take $O(n)$ time.

   (e) The second smallest element in a binary min-heap with all elements with distinct values will always be a child of the root.

2. (20 pts) This exercise is to learn binary search tree operations

   (a) Draw the sequence of binary search trees which results from inserting the following values in left-to-right order, assuming no balancing. 15, 10, 31, 25, 34, 56, 78, 12, 14, 13

   (b) Starting from the tree at the end of the previous part, draw the sequence that results from deleting the following nodes in left-to-right order: 15, 31, 12, 14.

   (c) After deleting them Draw the sequence of reinserting left-to-right in reverse order: 14, 12, 31, 15. in order into the tree and comment on the result?

3. (20 pts) Reading CRLS Chapter 6 and do the written

   (a) Exercises: 6.1-3, 6.1-4, 6.1-6, and 6.3-3

   (Note chapter 6 give a background to the coding exercise to use an array for a Max Heap. Also there is of course a nice Wikipedia article to look at https://en.wikipedia.org/wiki/Heapsort.)

   (b) Exercises: 12.3-2, 12.3-3, 12.3-4, B.5-4

   (Note that the degree of in an undirected graph (or tree) is the number links incident on the node. A leaf is a node with degree 1.)

Coding Exercise

4. (40pts) Implement a Max Heap for n elements as and array `int HeapArray[n+1];` of $n+1$ setting elements by placing the integers setting `HeapArray[0] = n` and copying the elements putting the elements in sequence into `HeapArray[i],   for i = 1,2,..., n` (May choose to have an longer array with extra space and a save a value `heapSize` to tell how many are in the heap. This is a useful index in any case!

   (a) Provide the in `HW4_codes/heap.cpp` to enable:

   ```
   (1) Insert random sequence to Heap array
   (2) Bottom up Heapify for Max Heap
   (3) Delete any key and restore Max Heap
   (4) Insert new key and restore Max Heap
   (4) Sort in place and print out array
   ```

   Put final code with `Makefile` in `/projectnb/alg504/username/HW4`

   (b) Analize the behavior of this code with the following plots:

   ```
   (1) Plot timing for range of sise n = 8, 16, 32,....2^20
   (2)  Histogram the performance over random permutation of UnsortedList100.txt
   (3) Combine these two part to define the average for
   n = 8, 16, 32,....2^20 and fit T(n) = a + b n Log[n] + c n*n
   ```

   Place these figures in `/projectnb/alg504/username/HW4` as well.