CSE 331 Midterm

Time: Oct 19, 2017, in class (80 mins).

Note: for simplicity, the base of “log” is always 2.

1. (5 + 5 + 5 = 15 points) Compute the growth rate for \( T(n) \). Assume that when \( n \leq 2 \), \( T(n) = 2 \). And when \( n > 2 \):
   • \( T(n) = 3T(n/2) + \sqrt{n + 3} = \Theta(n^{\log_2 3}) \);
   • \( T(n) = 4T(n/10) + n^{1/4} = \Theta(n^{\log_{10} 4}) \);
   • \( T(n) = (T(n/2))^2 = (T(n/2^{\log n}))^{2^{\log n}} = 2^{2^{\log n}} = \Theta(2^n) \).

2. (8 + 10 = 18 points) Prove (1) \( \sum_{i=1}^{n} i \log i = \Theta(n^2 \log n) \); (2) \( \sum_{i=1}^{n} (n - i) \times 2^i = \Theta(2^n) \).
   
   (1) First, it is easy to know \( \sum_{i=1}^{n} i \log i < n \times n \log n = O(n^2 \log n) \). Second, \( \sum_{i=1}^{n} i \log i > \sum_{i=n/2+1}^{n} \frac{n}{2} \log \frac{n}{2} = (\frac{n}{2})^2 \log \frac{n}{2} = \Theta(n^2 \log n) \). Thus, \( \sum_{i=1}^{n} i \log i = \Theta(n^2 \log n) \).
   
   (2) See the proof for linear time complexity of building heap.

3. (10 + 10 = 20 points) Let an array \( A = [7, 11, 10, 12, 0, 3, 2] \). Please build (1) MinHeap and (2) Max-Heap for \( A \). You can just show their visualization trees (it is better to show the building process as well, i.e., the change of the visualization trees step by step, to guarantee getting partial credits if your final answer is wrong). See Figure 1.

![Figure 1](image1.png)

4. (8 points) Is the following tree (Fig. 2) an AVL-tree? If not, please convert it to an AVL-tree.

5. (10 points) Given a sequence of input integers 11, 2, 3, 9, 6, 7, 8, please build the AVL-tree. Similar to problem 3, it is better to show the building process. See Figure 3.

6. (8 + 10 = 18 points) Suppose we want to insert \{12, 16, 13, 27, 5, 34, 1, 38, 49\} into a chaining hash table with size \( m = 11 \). Please show the final hash tables by the following two hash functions separately:
1. \( h(x) = x \mod 11; \)
2. \( h(x) = ((3x + 2) \mod 23) \mod 11. \)

See figure 4.

7. (6 + 5 = 11 points) (1) What are the time complexities of insertion sort, merge sort, and heap sort for the worst cases (suppose there are \( n \) integers)? (2) We know that AVL-tree requires that for each node, the difference of the heights of its two children should be at most 1. If we relax this difference to be at most 2, is the resulting tree’s height still \( \Theta(\log n) \) (suppose there are \( n \) nodes in the tree)? If yes, please prove it.

(1) \( \Theta(n^2) \), \( \Theta(n \log n) \), \( \Theta(n \log n) \).

(2) Yes. Use the same idea for proving the height of AVL-tree. Let the resulting tree have height \( h \), and \( \#(h) \) be the minimum possible number of nodes for all the height \( h \) trees which satisfy the modified requirement. Then we have \( \#(h) \geq \#(h-1) + \#(h-3) + 1 > 2\#(h-3) > 2^2\#(h-6) > \cdots > 2^{h/3} \).

Meanwhile, we know \( n \geq \#(h) \). Thus, \( n > 2^{h/3} \), that is, \( h < 3 \log n \). Since it is a binary tree, we always have \( h \geq \log n \). In total, \( h = \Theta(\log n) \).
Figure 4