PROBLEM SET 1, WRITTEN PART
Sorting and Balanced Binary Search Trees

Due: 11:55pm Thursday, October 4, 2018

---

1 Lower bounds for sorting (10 marks)

Consider a comparison-based sorting algorithm for sorting an input sequence of n numbers $(x_1, x_2, \ldots, x_n)$. Suppose that the sorting algorithm has the following additional information about the sequence to be sorted. The input sequence is a concatenation of n/k subsequences, each of which contains k elements. The sorting algorithm somehow knows that all the elements in each subsequence occur before all the elements in any later subsequence. Consequently, after each subsequence is sorted, the concatenation of all the sorted subsequences produces a sorted sequence. Using a decision tree argument, show that the lower bound on the number of comparisons needed to sort this sequence is $\Omega(n \log k)$.

2 Quickselect with median-of-medians pivots (10 marks)

In class, we saw a worst-case analysis of Quickselect using median of medians pivots. This pivot was the median of those medians taken from $\frac{n}{5}$ groups (each group being of size 5), and the resulting runtime was $O(n)$. Show that if we instead use $\frac{n}{9}$ groups of size 9, then the resulting runtime is still $O(n)$. To analyze the recurrence relationship, you may use the stack of bricks recursion, the substitution method, or another proof technique of your own choosing.

3 Insertion in 2-3 trees (10 marks)

Starting from an initially empty 2-3 tree, show the construction of the 2-3 tree by inserting the following keys in the specified order: 6, 9, 14, 15, 13, 2, 7, 18, 17.

For each insertion, show the stages of the tree after the initial insertion (which might lead to the creation of a temporary 4-node) and the subsequent operations that once again lead to a 2-3 tree.

4 Relaxed AVL trees (10 marks)

In class, we saw that AVL trees satisfy the height-balance property, which means that for every internal node, the left child subtree and the right child subtree have height differing by at most 1. Consider a slight relaxation of the height-balance requirement, where, for every internal node, the left child subtree and the right child subtree have height which differs by at most 2. Let's call these trees "Relaxed AVL trees".

Define $N(h)$ to be the minimum number of elements that can be stored in a Relaxed AVL tree. Prove that $N(h) = \Omega(k^h)$ for some constant $k > 1$. Note that you do not have to find the largest k for which this holds.