

CSCI 539 Algorithms

Homework 3

Due: October 25, 2001

1. Insert 2, 9, 4, 1, 7, 10, 3, 6, 5, 8 one by one into an initially empty AVL tree. Show the AVL tree after each insertion.
2. Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 44, 25 into a hash table of size $b = 11$ using closed hashing with the hash function $h(x) = x \bmod 11$. Complete the following hash table to show the final result of inserting these keys in the order given using double hashing with $h'(x) = (x \bmod 10) + 1$.

| | | | | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|---|---|----|
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $HT[i]$ | | | | | | | | | | | |

3. Can both `insert` and `findMin` (not `deleteMin`) be implemented in constant time $O(1)$?
4. A min-max heap is a data structure that supports both `DeleteMin` and `DeleteMax` in $O(\log n)$ per operation. The structure is identical to a regular heap, but the heap property is that for any node x at even depth, the key in x is the smallest in its subtree, and for any node x at odd depth, the key in x is the largest in its subtree. Assume that the root is at even depth of 0. For example, array 6, 81, 87, 14, 17, 12, 28, 71, 25, 80, 20, 52, 78, 31, 42, 31, 59, 16, 24, 79, 63, 18, 19, 32, 13, 15, 48 represents a min-max heap.
 - (a) How do you find the minimum in a min-max heap?
 - (b) Describe how `DeleteMin` can be done in $O(\log n)$ time.
 - (c) How do you find the maximum in a min-max heap?
 - (d) Describe how `DeleteMax` can be done in $O(\log n)$ time.