DS winter 2016 midterm retake

Sherri Shulman
DS Winter 2016 midterm retake

Feb 23, 2016

This exam is open book, open note, your can use the syllabus (and so the lecture notes) but no web surfing. 2 hours. [Note: there are a lot of trees here. When I say "Draw the intermediate trees", you don't have to draw *every* intermediate tree, but a sample to demonstrate whatever rotations are being used in the indicated data structure.]

1. (17 pts) *AVL trees* Start with an empty AVL search tree and insert the following keys in the given order: $15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1$. Show your tree after each insertion, and identify what rotations you used, if any. Label all nodes with their balance factors.

2. (17 pts) *Splay trees* Start with an empty splay tree and insert the following keys in the given order: $20, 10, 5, 30, 40, 25, 8, 35, 7, 23$. Show your tree after each insertion. Do bottom up splaying.

3. (17 pts) *B-trees* Start with an empty Btree of order 3 and insert the keys: $2, 1, 5, 6, 7, 4, 3, 8, 9, 10, 11$ in that order. Draw the Btree after each insertion.

4. (17 pts) *Hash tables* Use linear probing, a hash table with $b = 17$ buckets, and the hash function $f(k) = k \mod b$. Start with an empty hash table and insert elements whose keys are $7, 42, 25, 73, 14, 38, 8, 22, 34, 11$. The elements are inserted in this order.

   (a) Draw the hash table following each insert.

   (b) What is the loading factor of your table after the last insert?

   (c) What is the maximum and the average number of buckets examined in an unsuccessful search of your table?

   (d) What is the maximum and the average number of buckets examined in a successful search of your table?

5. (17 pts) *Heaps* Consider the array $theHeap = [-, 30, 17, 20, 15, 10, 12, 5, 7, 8, 5, 2, 9]$

   (a) Heapify the tree using the linear time algorithm (buildheap and percolatedown, figure 6.14 and 6.12). Make this a minheap.

   (b) Insert the elements $1, 10, 6, 4$ in that order

   (c) Perform 3 remove min operations.

6. (15 pts) Suppose you have a heap represented as an array as described in the textbook. Give an $O(1)$ function that determines the height of the heap.