CS130A Homework 2

Due 11:59 AM 2019-07-24 (end of discussion section)

1. A d-heap is a generalization of a binheap, but instead of storing a logical binary tree in an array, each node can have d children. A d-heap can still be stored in a densely packed array.
    a. For an entry located in position i, where are the parents and the children? Sate if this is zero based or not.
    b. Create a 5-heap of height 2 (provide the array representation). Draw the aftermath of calling deleteMin (in the finished state).
2. Leftist Heaps
    a. Insert the elements 1, 2, 3, 4, 5 into a leftist heap in sequence. Draw the resulting tree structure after each element insertion.
    b. Insert the elements 8, 7, 6, 10, 9 into a second leftist heap. Draw the resulting tree structure at the end.
    c. Merge the heaps from a and b. Draw the tree structure at the end.
3. AVL Trees
    a. Insert the values 10 20 30 40 50 60 70 into an AVL tree.
    b. Insert the values 140 100 110 130 120 80 90 into the same AVL tree and draw the result
    c. Delete nodes 10 50 and 100 and draw the result
    d. In an AVL tree, if the path-length from the root to the left-most child is 2, what is the longest path-length to the rightmost child?
4. Red-black Trees
    a. Insert the values 10 20 30 40 50 60 70 into a red-black tree and draw the result
    b. Insert the values 140 100 110 130 120 80 90 into the same red-black tree and draw the result
    c. Delete nodes 10 50 and 100 and draw the result
    d. What is the maximum number of nodes in a red-black tree of black-height 4?
    e. What is the minimum number of nodes in a red-black tree of black-hight 4?
5. Amortized Analysis
    a. In your programming assignment, you implemented a binary heap as an array or vector. Suppose you used an array. When the heap filled, you doubled the size of the array and copied the elements into the new array. Assume that copying n elements is O(n). Starting with an initial array size of 1, what is the amortized cost of inserting M elements into your heap? Note, the costs for a fixed array implementation are well described in your book.
    b. Assume you have a hash function family that magically does not collide until the load factor on your array implementation reaches 50%, even though your table size is always a power of 2. Every time you resize your table, you must rehash each element. What is the amortized cost of insertions?
    c. Using the binary counter example from the slides, what is the amortized cost of increments if the cost to flip the k-th bit (0000 to 0001 flips bit 1) is k?

    d. Using the binary counter example from the slides, what is the amortized cost of increments if the cost to flip the k-th bit (0000 to 0001 flips bit 1) is 2k?
    e. Using the binary counter example from the slides, what is the amortized cost of increments if the cost to flip the k-th bit (0000 to 0001 flips bit 1) is k^2?
6. Minimum Spanning Trees
    a. Prove that if all of the edge costs in graph are unique, the minimum spanning tree is unique
    b. True or False: If the edge costs of a graph double, the cost of the minimum spanning tree must also double
7. Shortest Path Algorithms
    a. True or False: If the edge costs of a complete graph double, the cost of the shortest path between two nodes must also double
    b. True or False: If the edge costs of a complete graph are unique, the path for the shortest path between two nodes must be unique
    a. Construct an undirected graph of 5 or more vertices that contains a negative cycle for which there exists a shortest path that does not contain the negative cycle.
8. Using the B-tree definition in wikipedia (https://en.wikipedia.org/wiki/B-tree), and more specifically, the 2-3-4 tree definition (https://en.wikipedia.org/wiki/2-3-4_tree) as a B-tree of order 4, answer the following:
    a. What is the minimum number of values stored in a 2-3-4 tree of height 1?
    b. What is the maximum number of values stored in a 2-3-4 tree of height 1?
    c. How many nodes are in the tree in b?
9. Using the B+ tree definition in wikipedia (https://en.wikipedia.org/wiki/B%2B_tree), answer the following:
    a. On my home system, if I run `sudo blockdev --getbsz /dev/mapper/pdc_ghcifidie1`, I get 4096. This means my block size is 4096 bytes. If a particular key is 8 bytes, what is the most efficient choice of b for a b-order B+ tree? State your reasons for any padding.
    b. What is the maximum number of int keys for the tree in part a with 3 levels of index?