PAL labs 11

November 30, 2022

A skip list is originally empty ant then the keys are inserted into the list one by one in the given order. The numbers in the parentheses denote the level of the corresponding node given by the number of coin tosses. Each time, the coin is being tossed until it comes up tails. Draw the resulting list.
16(3) 23(2) 18(2) 5(2) 15(1) 19(1) 33(1) 11(2) 21(2) 4(1) 22(2) 6(2) 17(4) 10(1) 9(1) 28(4)

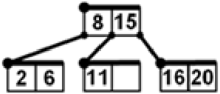
There are two skip lists of length N. Describe an effective algorithm which merges these two skip lists into one skip list of length $2N$. What is the asymptotic complexity of your solution?

Is it possible to reverse the order of the keys in a skip list in time which is asymptotically less than $\Theta(N \times log(N))$?

Describe how to use a skip list as a priority queue. Compare the complexity of operations Insert and ExtractMin to the complexity of the same operations in a binary heap. Which of these two data structures performs asymptotically better?

There is a discussion about the sum of level values of all nodes in a skip list of length N. Professor Highman says that the sum is on average proportional to $N \times log(N)$. Professor Lowman says that the sum is on average proportional to just N. Professor Middleman says that both cases might happen and that it depends on the data. Decide whether any of the three professors is right.

Insert keys 14 and 10 into the left B-tree and insert keys 7, 5 into the right B-tree. What will be the contents of the root after the insertion?



Two empty B-trees of order 1 (max 2 keys in a node) are isomorphic. Let T1 and T2 be two unempty B-trees with the respective roots R1 and R2. T1 and T2 are isomorphic iff both 1. and 2. holds:

1. The root of T1 contains the same number of keys as the root of T2.
2. The left subtree of R1 is isomorphic to the left subtree of R2, the right subtree of R1 is isomorphic to the rightsubtree of R2 and the middle subtree (if it exists) of R1 is isomorphic to the middle subtree of R2.

What is the number of non-isomorphic B-trees with A) 0, B) 1, C) 3, D) 4, E) 7 nodes?

Extend your solution to the previous problem and find a general recursive formula which specifies the number of non-isomorphic B-trees of order 1 for any given number of nodes in the tree.

We are given a) B-tree b) B+ tree. The order of the tree is 10 and it contains exactly 100 000 keys. What is the maximum and minimum possible height of the tree? What is the maximum and minimum possible number of keys in the tree?

Suppose that a B-tree of order 1 is originally empty. Insert, in the given order, the given keys into the tree
25, 13, 37, 32, 40, 20, 22.
Draw the tree after each insertion. Next, delete the keys from the resulting B-tree in the order:
13, 25, 40, 22, 20, 37, 32.
Draw the tree after each deletion.

Describe an effective operation IncreaseKey and DecreaseKey in a skip list. The amount by which the key value is increased/decreased will be the parameter of the operation.