Assignment 17, Data-Structures

Oleg Sivokon

<2016-06-18 Sat>

Contents

1 Problems

1.1 Problem 1

Given integer n, suggest data-structure S with following properties:

1. `build` takes $O(n \lg n)$ time.
2. `insert` takes $O(\lg n + d)$ time.
3. `delete` takes $O(\lg n + d)$ time.
4. `successor` takes $O(1)$ time.

There was a descritption of what d is, but it uses some runic Hebrew, and is impossible to decypher. I'll ignore it.

1.1.1 Answer 1

1.2 Problem 2

Does there exist a red-black tree s.t.:

1. It contains 3 black and 4 red nodes.
2. It contains 4 black and 3 red nodes.
3. It contains 5 black and 2 red nodes.
4. It contains 6 black and 1 red node.

1.2.1 Answer 2

Yes, this configuration is possible. Consider root black node with two black children. Each child of the second level has only red nodes. In this tree all simple paths have the same number of black nodes. All red nodes have black children (leafs).

1.2.2 Answer 3

Such tree is impossible. Since the root node must be black, we should have to place 3 remaining black nodes on two paths from the root to the leafs, but we cannot divide 3 evenly in two paths. And since we cannot, we would have to violate the tree property which requires the same number of black nodes on all simple paths from root to be the same.

1.2.3 Answer 4

Yes, such tree is possible: black root node with two red children and four black grandchildren. Each simple path from root has two black nodes. No red node is a parent of another red node.

1.2.4 Answer 5

No, such tree is impossible. The number of black nodes in a red-black tree must be odd (the root and two sub-trees with equal number of black nodes).

1.3 Problem 3

Suggest data-structure which allows the following operations:

1. `insert` in time $O(\lg n)$.

2. `delete` in time $O(\lg n)$.
3. `pair-diff` *(find two numbers which differ precisely by given amount)* in time $O(n)$.
4. `sum` *(sums all numbers within given range)* in time $O(\lg n)$.

1.3.1 Answer 6

1.4 Problem 4

Suggest data-structure which allows the following operations:

1. `search` in time $O(\lg n)$.
2. `insert` in time $O(\lg n)$.
3. `delete` in time $O(\lg n)$.
4. `mode` in time $O(1)$.
5. `mark` *(insetion time)* in $O(\lg n)$.

1.4.1 Answer 7