Feedback — Balanced Search Trees

You submitted this quiz on **Fri 26 Feb 2016 11:31 AM PST**. You got a score of **3.00** out of **3.00**.

To specify an array or sequence of values in an answer, separate the values in the sequence by whitespace. For example, if the question asks for the first ten powers of two (starting at 1), then the following answer is acceptable:

```
1 2 4 8 16 32 64 128 256 512
```

If you wish to discuss a particular question and answer in the forums, please post the entire question and answer, including the seed (which can be used by the course staff to uniquely identify the question) and the explanation (which contains the correct answer).

Question 1

(seed = 988592)
Consider the left-leaning red-black BST whose level-order traversal is:

```
79 43 95 33 76 94 97 20 42 71 77 16 70
```

List (in ascending order) the keys in the red nodes. A node is red if the link from its parent is red. Your answer should be a sequence of integers, separated by whitespace.

You entered:

16 43 70

Your Answer		Score	Explanation
16 43 70	✔	1.00	
Total		1.00 / 1.00	

Question Explanation

The correct answer is: 16 43 70

The shape of a BST is uniquely determined by its level-order traversal.

To deduce which links are red, recall that the length of every path from the root to a null link has the same number of black links; apply this property starting from nodes at the bottom.

Question 2

```
(seed = 397568)
Consider the left-leaning red-black BST whose level-order traversal is

    32 18 93 14 31 51 95 46 58 42          ( red links = 42 51 )

What is the level-order traversal of the red-black BST that results after
inserting the following sequence of keys:

    28 97 26

Your answer should be a sequence of 13 integers, separated by whitespace.
```

You entered:

```
32 28 93 18 31 51 97 14 26 46 58 95 42
```

Your Answer		Score	Explanation
32 28 93 18 31 51 97 14 26 46 58 95 42	✔	1.00	
Total		1.00 / 1.00	

Question Explanation

```
The correct answer is: 32 28 93 18 31 51 97 14 26 46 58 95 42

Here is the level-order traversal of the red-black BST after each insertion:

    32 18 93 14 31 51 95 46 58 42          ( red links = 42 51 )
28: 32 18 93 14 31 51 95 28 46 58 42       ( red links = 28 42 51 )
97: 32 18 93 14 31 51 97 28 46 58 95 42    ( red links = 28 42 51 95 )
26: 32 28 93 18 31 51 97 14 26 46 58 95 42 ( red links = 18 42 51 95 )
```

Question 3

```
(seed = 34007)
Which of the following statements about balanced search trees are true? Check all that apply. Unless otherw
ise specified, assume that the 2-3 tree and red-black BSTs are as described in lecture (e.g., 2-3 trees are
perfectly balanced and red-black BST are left-leaning red-black BSTs with internal links colored either red
or black).
```

Your Answer	Score	Explanation

☐	✔	0.20	This is a tricky one. We show that the number of rotations is at most 3N. First we claim that the total number of color flips is at most N: Each insertion (other than the first one) colors one link red and then performs rotations and color flips to restore the red-black BST invariants; rotations do not change the number of red links; each color flip decreases the number of red links by 1 (typical case) or by 2 (if performed at the root). During an insert operation, after any two consecutive rotation operations, there is a color flip.
Consider inserting N distinct keys into an initially empty red-black BST. Then, in the worst case, the total number of rotations is linearithmic in N.			
☐	✔	0.20	If the link between x and its parent is red, then the black height of the subtree rooted at x will be one less than the black height of the subtree rooted at y.
Let x and y be two sibling nodes in a red-black BST. Then, the black height of the subtree rooted at x equals the black height of the subtree rooted at y.			

☑ It is possible to construct a red-black BST from a sorted array of N distinct keys in linear time.	✔	0.20	Create a perfectly balanced BST by choosing the (upper) median key to be the root and recursively construct a perfectly balanced BST in each subtree. The resulting tree can be (uniquely) colored to make it a red-black BST. One way to see how is to color every link in the bottommost level red so that the red-black BST has perfect black balance. Note that since we use the upper median (instead of the lower median), whenever there is a right-leaning link that is red, the sibling left-leaning link will also be red. Now, repeatedly apply color flip operations until there are no more right-leaning red links.
☑ It is possible to implement the left (or right) rotation operations in a BST in constant time.	✔	0.20	A left or right rotation changes only a constant number of pointers.
☑ Consider inserting N keys in ascending order into an initially empty red-black BST. Then, after each insertion, the height of the tree either strictly increases or remains unchanged.	✔	0.20	The BST remains perfectly balanced - after the ith insertion, its height is exactly floor(lg i), which is a monotonically nodecreasing function of i.
Total		1.00 / 1.00	