Feedback — Problem Set-6

Help Center

You submitted this quiz on **Sat 21 Nov 2015 8:38 AM PST**. You got a score of **2.00** out of **5.00**. You can attempt again in 1 minutes.

Question 1

Suppose we use a hash function h to hash n distinct keys into an array T of length m. Assuming simple uniform hashing --- that is, with each key mapped independently and uniformly to a random bucket --- what is the expected number of keys that get mapped to the first bucket? More precisely, what is the expected cardinality of the set $\{k : h(k) = 1\}$.

Your Answer		Score	Explanation
$m/(2n)$			
$1/n$	✖	0.00	Use linearity of expectation, with one indicator variable for each key. Don't forget to sum over the keys.
n/m			
$n/(2m)$			
$1/m$			
m/n			
Total		0.00 / 1.00	

Question 2

You are given a binary tree (via a pointer to its root) with n nodes, which may or may not be a binary search tree. How much time is necessary and sufficient to check whether or not the tree

satisfies the search tree property?

Your Answer		Score	Explanation
$\Theta(n \log n)$			
$\Theta(\log n)$			
$\Theta(n)$	✔	1.00	For the lower bound, if there is a violation of the search tree property, you might need to examine all of the nodes to find it (in the worst case).
$\Theta(height)$			
Total		1.00 / 1.00	

Question 3

You are given a binary tree (via a pointer to its root) with n nodes. As in lecture, let size(x) denote the number of nodes in the subtree rooted at the node x. How much time is necessary and sufficient to compute size(x) for every node x of the tree?

Your Answer		Score	Explanation
$\Theta(n)$			
$\Theta(n^2)$	✖	0.00	Can you do better?
$\Theta(height)$			
$\Theta(n \log n)$			
Total		0.00 / 1.00	

Question 4

Which of the following is *not* a property that you expect a well-designed hash function to have?

Your Answer		Score	Explanation
○ The hash function should "spread out" most (i.e., "non-pathological") data sets (across the buckets/slots of the hash table).			
◉ The hash function should "spread out" every data set (across the buckets/slots of the hash table).	✔	1.00	As discussed in lecture, unfortunately, there is no such hash function.
○ The hash function should be easy to compute (constant time or close to it).			
○ The hash function should be easy to store (constant space or close to it).			
Total		1.00 / 1.00	

Question 5

Suppose we relax the third invariant of red-black trees to the property that there are no *three* reds in a row. That is, if a node and its parent are both red, then both of its children must be black. Call these *relaxed* red-black trees. Which of the following statements is *not* true?

Your Answer	Score	Explanation
○ The height of every relaxed red-black tree with n nodes is $O(\log n)$.		
○ There is a relaxed red-black tree that is not also a red-black tree.		

◉ Every red-black tree is also a relaxed red-black tree.	✖	0.00	The third invariant is only easier to satisfy, and the other three invariants are the same.
○ Every binary search tree can be turned into a relaxed red-black tree (via some coloring of the nodes as black or red).			
Total		0.00 / 1.00	