Algorithms I

Tutorial 1

August 12, 2016

Problem 1

CLRS problem 2-1

Problem 2

Arrange the following list of functions and arrange them in ascending order of growth rate. i.e. If $g(n)$ immediately follows $f(n)$, then it should be the case that $f(n)$ is $O(g(n))$:

- $f_1(n) = 10^n$
- $f_2(n) = n^{\frac{1}{3}}$
- $f_3(n) = n^n$
- $f_4(n) = \log_2 n$
- $f_5(n) = 2^{\sqrt{\log_2 n}}$

Problem 3

Find a tight upper bound for the running time of the following code in terms of n.

```
for (i = 1; i <= n; i += 1) {
    for (j = i; j <= n; j += i) {
        x = 5;
    }
}
```

Problem 4

Given a sorted sequence of n distinct integers $a[1], a[2], a[3] \ldots a[n]$, find if there exists an index i such that $a[i] = i$.

Problem 5

A cyclically sorted array is the array obtained by sorting an array and then doing a cyclic rotation. e.g.$[1, 2, 3], [3, 1, 2], [2, 3, 1]$ are cyclically sorted arrays. Your task is to find an element in such an array in $O(\log n)$, n is the number of elements in the array.

Problem 6

Assume you have stacks and queues that can store any data type. Do a pre-order traversal of a binary tree without recursion.

Problem 7

With the same assumption as previous problem, do an in-order traversal of a binary tree without recursion.

Problem 8

Check if two given binary trees are same. (We call two binary trees same if they are structurally identical and the corresponding nodes have same value).

Problem 9

You are constructing a balanced BST from an array of n elements. You are allowed to store extra data in each node. Now, you need to find the median element in the BST in $O(\log n)$. If n is even, choose the larger of the two median elements.

Problem 10

Find the second minimum key in a balanced BST in $O(\log n)$ without using *insert* or *delete* operation.

Problem 11

Check whether a given binary tree is a BST in $O(n)$ time with $O(1)$ additional space (excluding recursion stack).

Problem 12

Is the operation of deletion commutative in the sense that deleting x and then y from a binary search tree leaves the same tree as deleting y and then x? Argue why it is or give a counterexample.

Problem 13

What is the largest possible number of internal nodes in a red-black tree with black-height k? What is the smallest possible number?

Problem 14

Suppose that you have n values coming one by one. You are asked to find the k^{th} smallest value after each input. If there are less than k values, you do not need to do anything. Your algorithm should run in time $O(n \log k)$ overall.

Problem 15

Suppose that you have values coming one by one, you are asked to find the median of the new set of values. The insert can take $O(\log n)$, but you should be able to find the median in $O(1)$. How will you do this?