Problem Solving Workshop #5 Tech Interviews and Competitive Programming Meetup

April 24, 2016 https://www.meetup.com/tech-interviews-and-competitive-programming/

Instructor: Eugene Yarovoi (can be contacted through the group Meetup page above under Organizers)

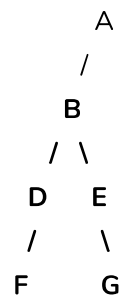
More practice questions: leetcode.com, glassdoor.com, geeksforgeeks.org
Books: Elements of Programming Interviews, Cracking the Coding Interview
Have questions you want answered? Contact the instructor, or ask on Quora. You can post questions and follow the instructor and other people who write about algorithms.

Try to find optimized solutions, and provide a time and space complexity analysis with every solution for the algorithms questions.

Easy Problem

Given a binary tree, find the longest path in the tree (treating the edges as being undirected).

Example Input / Output:



The nodes in bold form the longest path: F -> D -> B -> E -> G

Medium-Hard Problem #1

You're given a string S, and a list L of pairs where each pair (i, j) is a rule that indicates that you may at any time swap the character at position i in S with the character at position j in S. Generate the lexicographically largest (last in dictionary order) string that you can by swapping characters. The same (i, j) pair may be reused any number of times to perform as many swaps as needed.

Example Input:
S = "abdc"
L = [(0,3), (2,3)]

Output: dbca

Explanation: It is possible to get the strings abcd, abdc, cdba, cbad, dbac, dbca. You should print only dbca which is lexicographically largest (last in dictionary order) out of these. You can get it by starting with abdc, using the (2,3) rule to swap positions 2 and 3 to produce abcd, and then using the (0,3) rule to produce dbca.
dcba is even larger lexicographically, but it's not possible to obtain it by swaps (quite obviously since there are no rules that can move anything from position 1).

Medium-Hard Problem #2

There is an MxN grid (matrix) where each cell contains a value (may be positive, zero, or negative). When you visit a cell, you collect the value in that cell. You start at (0,0), the upper left corner. The coordinate system is image coordinates: x increases going right, and y increases going down.

If you are located at position (a, b), you may jump next to any position (a+dx, b+dy) where dx and dy are integers greater than or equal to 1, and (a+dx, b+dy) is in-bounds of the grid. In other words, every jump is required to move you to the right and downwards by at least (but perhaps more than) one row and one column.

You can jump as many times as you want, and there is no specific cell where you have to end up. Determine the maximum sum of values you can obtain.

Example Input:

0	-10	-10	-10
5	-10	6	-10
15	7	-10	-10
-10	-10	-10	20

Output: 27
Explanation: We start in the cell labeled 0. The best sequence of jumps is to first jump to 7 and then to 20. We can't jump to 5 or 15 because the x-displacement of the jump would be 0, but that's not allowed. We could jump to 6, but we then we couldn't jump to 7 (since jumps must have a positive x-displacement only), and 6 + 20 would be less than 7 + 20. We can't jump to 6 after jumping to 7 either, because that jump would have a negative y-displacement.

Design Problem

Imagine you are Twitter. How would you design the “top 100 trending hashtags” feature, if it were not already designed? There is deliberate ambiguity in this problem -- you will need to make reasonable assumptions. When coming up with your system design, you will need to modify / add to the design of one or more existing Twitter systems, so you will need to make reasonable assumptions for what the existing architecture looks like.