Problem Solving Workshop #15 Tech Interviews and Competitive Programming Meetup

November 5, 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.

Coding Problem

Givena string S where all the characters are either '(' or ')', find the longest contiguous substring that is a valid parenthezation. For example, "(())" is valid, but "()))((" is not.

Example Input: S = "))**(())**)"
Output: "(())"
Explanation: the longest substring of S that is a valid parenthezation is the substring from index 2 (0-indexed) to index 5, inclusive on both ends.

Algorithm Problem #1

Given a list L of strings, find two strings s_1 and s_2 in L such that s_1s_2 (the concatenation of s_1 and s_2) is a palindrome of maximum length.

Example Input: L = ["abcba", "abcd", "cba"]
Output: "abcd", "cba"
Explanation: When you concatenate those two words, you get "abcdcba", which is a palindrome. No other choice of two words from L could have produced a longer palindrome when concatenated.
Complexity guidance: analyze the complexity of your algorithm in terms of N, the number of strings in L, and W, the average length of a word. **(Easy)** $O(N^2W)$, **(Easy-Medium)** $O(NW^2)$, **(Hard)** O(NW) = O(size of input).

Algorithm Problem #2

You have N mines in a minefield that need to be cleared by futuristic robots. When a robot finds a mine, it steps on it, destroying the mine and the robot.

The trouble is, you only have one robot, and potentially more than one mine. Luckily, the robot can also build other robots. It takes a robot T units of time to build another robot, during which it can't do anything

else. The new robot is identical in every respect to the original, including the ability to build more robots in turn. You have unlimited raw materials with which more robots can be constructed.

For each mine i, you have some value M_i that denotes how long it will take for a robot to find and destroy that mine (there's some search process involved). When you send a robot to destroy a mine, the command has to be completed all the way (robot cannot be recalled). The robot can't do anything else while it's finding the mine, and is destroyed together with the mine when the command is complete. Multiple robots can search for different mines at the same time, but two robots can't search for the same mine because they'd get in each other's way.

Given the integer T that denotes how long it takes to build a robot, and a list M of mine search times (integers), find the minimum amount of time needed to clear the minefield.

Example Input: M = [2, 9, 2, 2], T = 3
Output: 12
Explanation: At time 0, we command the robot to buid another robot, so at time 3 we have 2 robots. We command one of them to find the "9" mine, which will finish at time 12 (current time = 3, mine time = 9, 3+9 = 12). We command the other robot to build a robot, so we have 2 available robots at time 6. We command both of them to build more robots. At time 9 we have 4 available robots, and we send 3 of them to get each of the "2" mines, which get destroyed at time 11. The 9 mine is clear at time 12 , and so the whole process takes 12 units of time.