Problem Solving Workshop #7 Tech Interviews and Competitive Programming Meetup

May 14, 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.

Coding Problem (easy-medium)

Given an array A, count the number of contiguous subarrays that sum to a target number T.

Example Input: A = [4, 6, 3, 7, -10], T = 10
Output: 3
Explanation: [4, 6], [3, 7], and [4, 6, 3, 7, -10] are contiguous subarrays that sum to 10.

Algorithm Problem

Consider the parentheses-balance problem, the problem of determining whether a string of parentheses is well-formed, or in other words, every opening parenthesis can be matched to a closing parenthesis that comes after it, and there are no extra closing parentheses.

Example Input: “((()))()”
Output: yes
Explanation:

```
(        (          (          )        )        )      (      )
|        |           |         |        |        |      |      |
|        |           ----------         |        |       ------
|         ------------------------------         |
--------------------------------------------------
```

(i) **[Easy]** Give an O(N) algorithm to decide whether a string of paretheses is well-formed.
(ii) **[Medium]** Suppose you now have K threads. How would you parallelize your solution? Your target running time should be O(N / K) work per thread, assuming that K is not too large (you may have some other small factor in the complexity related to K).

(iii) **[Medium-Hard]** Suppose you now have unlimited processors and threads. Your ability to use parallelism is limited only by the fact that it takes some (assume O(1)) amount of time to start a thread, and pass arguments to it. Can you improve your previous solution?

System Design Problem (medium to hard)

The interviewer states the problem like this: Consider an existing large website where there are lots of images submitted by users. For example, Wikipedia, Pinterest, DeviantArt. How would you design a web crawler to crawl all pages belonging to that domain, and download all the images?

At this point, you would need to ask clarifying questions before you can attempt the problem. Imagine the conversation goes as follows:

You: How large is the website and how many images do you estimate there are?
Interviewer: The site may have on the order of 1 billion images.
You: Each image may be 100KB-1MB, so we're talking hundreds of terabytes of data. I assume we want to download this data into some kind *distributed* database, since 1 machine can't hold this.
Interviewer: Yes. And no requirement to organize the images in any way, just download them all.
You: Can I assume I have the storage set up and an API call that adds 1 image to the storage?
Interviewer: Yes. Just design the crawler.
You: How do we know what pages are available in the domain? Is the website organized so that we will find all the pages by starting from the homepage for the domain and following links?
Interviewer: Yes. Also assume for the time being you don't have to deal with robots.txt or the like.
You: Can I assume a static page structure too, the link structure of the site isn't changing much?
Interviewer: Start with that assumption, you can think how to weaken this assumption later.
You: Can I assume images are referenced by img tags, and they're not embedded in some other medium like Flash?
Interviewer: Yes. The images will be easily accessible like that.
You: Because there's so much data, this process may take a long time.
Interviewer: Yes. Think about your design in the context of this huge scale.
You: And probably the websites will detect you're crawling them and block you.
Interviewer: True. Assume that doesn't happen, or that you can circumvent it.

At this point, there's still plenty of ambiguity, but assume reasonable answers of your choice for your remaining questions.

Design the system.