Problem Solving Workshop #38 Tech Interviews and Competitive Programming Meetup

August 26, 2018 https://www.meetup.com/tech-interviews-and-competitive-programming/

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

Graphs Special Edition

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.

Problem #1, "Airline Booking" (Easier)

There are a number of cities (N) and some number of one-way flights between them (e.g. a flight could be San Jose -> Phoenix). Each flight has a price associated with it. When a customer searches for a booking to get from an initial city S to a destination city D, you want to find the lowest-cost itinerary from S to D that gets there using the available flights. The total cost of the itinerary is the sum of the cost of all the flights. For example, if a customer flies from San Francisco -> Houston -> Boston, the cost of the trip will be the cost of San Francisco -> Houston plus the cost of Houston -> Boston. However, given that the customer is looking for a practical itinerary, you want to limit it to the lowest-cost itinerary that has 3 or fewer hops (makes 2 or fewer intermediate stops). In fact, design an algorithm that does what airline sites do: display the best 2-stop route, best 1-stop route, and best non-stop route.

Your input will be a list of 3-tuples (departure airport, destination airport, price) plus the customer's departure and arrival cities.

Example Input:
Available flights: [(SFO, PHX, 120), (PHX, BOS, 200)]
Departure: SFO
Arrival: BOS

Output: Best 1-stop: SFO->PHX->BOS, $320
Non-stop not available.

Explanation: there is no non-stop flight, and the best (only, in this simple case) 1-stop is SFO->PHX->BOS. Its cost is 120+200 = 320. There are also no 2-stop flights so we can either not display it, or display it as unavailable.

Problem #2, "Distributed Debugging" (Easier)

You have a distributed system. When a sequence of events happens, different parts of the distributed system may see -- and log -- different subsets of those events. The events always appear in the same order in each component's log as the order in which they actually occurred, but each log may only have a subset of the events. Given the sequence recorded in each log, can you recover the original sequence (or say that there isn't enough information to do so)?

Example Input:
frontend server log: connected to server, successful login, logged out
authentication module log: failed login, successful login, logged out
security audit log: connected to server, failed login, initiated bank transfer
feature usage log: successful login, initiated bank transfer, logged out

Output: connected to server, failed login, successful login, initiated bank transfer, logged out

Explanation: the output shown is the only sequence of events consistent with the components seeing the events in the order shown in the input, given the problem's condition that each component always sees events in the order that they occur.

Problem #3, "Crocodiles" (Harder)

You are in a castle having N rooms numbered from `0` to `N-1`, and M bidirectional passageways between rooms. One of the rooms is your "starting room", and some subset of the rooms are "exit rooms" that lead to an exit from the castle. The castle is the lair of an evil mastermind who had you trapped there, and you're trying to escape as quickly as possible. It takes you one unit of time to use a passageway to cross between the two rooms it connects, and if you are in an exit room, you can escape immediately (0 units of time).

(a) What is the shortest time in which you can escape? (Note: at this point, this problem is a slight variation on a classic problem. This is just a warmup.)

Now, the evil mastermind happens to have K pet crocodiles. Every time you are inside a room, the mastermind will choose up to K passageways into which to place a crocodile (the mastermind may rearrange the crocodiles every time you enter a room; that is, every time you move). Passageways containing crocodiles cannot be crossed, and the mastermind hopes to delay your escape as much as possible by placing them.

Assuming the mastermind will play an **optimal** strategy to delay you as much as possible, what is the best time in which you are **guaranteed** to be able to escape (or it may be impossible -- in that case, you must state so)?

The input consists of the value N (number of rooms), a list of all the passageways (each passageway specified as two room numbers representing the rooms it connects), the number of your starting room S, the set of exit room numbers, and the parameter K (number of crocodiles).

(b) Solve this version of the problem.
(c) What if the passageways have a travel time associated with them (e.g. one may take 3 units of time and another take 5 units of time)? How would you modify your approach?

Problem #4, “Kitchen Recipes" (Harder)

You have a number of recipes for making certain food items from other food items. All the recipes take the form "if you have ingredients X1, X2, ... Xn, this recipe produces Y1, Y2, ...". For example, a recipe like [flour, milk, sugar, butter] -> [coffee cake, crumb topping] is a recipe that makes coffee cake and the crumb topping out of flour, milk, sugar, and butter. Any produced items can themselves be ingredients in other recipes. Note that there may be multiple recipes (with different ingredients) that produce the same thing; for example, there can be several ways to make cake.

Your kitchen is initially stocked with a certain set of ingredients, of which you have an infinite supply. You want to determine all the food you can make out of those ingredients by following the recipes. It may be the case that you make a food item by following a recipe, and then use that item as an ingredient in another recipe.

Example Input:
Initial ingredients: [almonds, eggs, sugar, milk]
Recipes:
[egg white, almond flour, sugar] -> [macaron]
[almonds] -> [almond flour]
[eggs] -> [egg white, egg yolk]
[eggs, flour, milk, sugar] -> [cake]

Output: You can make: almonds, eggs, sugar, milk, almond flour, egg white, egg yolk, macaron.

Explanation: you can make everything except cake by following some sequene of recipes. You can't make cake because you don't have flour and don't have any way to produce any (almond flour is a different ingredient). You can make egg white, egg yolk, and almond flour right away from the ingredients you start with. Then you can make macarons from the egg white and almond flour you made, combined with some sugar that you started with.

Problem #5, "Download ALL the things!" (Algorithmic Design)

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: I will set up some distributed data storage system. Can I assume I have that set up and an API call that adds 1 image to the storage once I have that image downloaded locally on a worker machine?
Interviewer: Yes. Just design the crawler. You should mention what data schema you will use to organize any data your crawler directly stores in a database, though. You don't have to worry about the database's internal implementation details, as this is a system built by someone else.
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. Don't focus your answer too much on this aspect. Let's get to the core logic first.

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

Guidance: For these kinds of problems involving distributed systems, first figure out how you would solve the problem at small scale. What if you have the entire domain downloaded locally on one machine and there's only a few pages? Then think how you will make it efficient enough for millions of pages. Then, try to make more and more subsystems into distributed systems.