Problem Solving Workshop #20 Tech Interviews and Competitive Programming Meetup
February 25, 2017 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: glassdoor.com, careercup.com, geeksforgeeks.org
Books: Elements of Programming Interviews, Cracking the Coding Interview
Have questions you want answered? Contact the instructor, or ask people on Quora. You can post questions and follow the instructor and other people who write about algorithms.

Generally speaking, in design questions, there's always lots of ambiguity, so you would start by asking clarifying questions to the interviewer. Since you don't have an interviewer here, assume what you think are reasonable answers to the questions you would ask.

Object-Oriented & System Design

Imagine you are member of Uber's founding team and you're developing a prototype for the Uber ride system. Create an object-oriented design for the system. Because this is a prototype, you don't have to deal with scalability or build a distributed system -- for now, this system will operate at small scale. First create the class design with purely in-memory objects holding all the data (define APIs for external sources such as driver GPS to send you data), and then consider how you would persist certain information in a database layer.

Make sure to consider the functionality that would be core to a minimum viable prototype of this service (possibly not super reliable due to not being a distributed system).

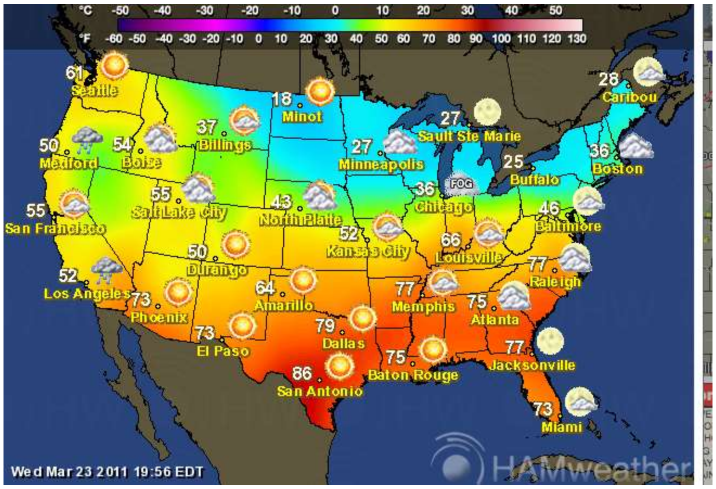
Feature Design

Consider a multi-user filesystem, such as a code repository, that can be accessed through a web browser. Suppose there is a box into which the user can type the path they would like to access, e.g. trunk/java/com/me/toplevel/package/AlgorithmSolverMuchWow.java.

Design a feature in which the site will propose path autocompletions to you as you're typing, based on the path you've already typed so far. At a minimum, the system should take into account how likely you are to be looking for a particular file given the path you've already typed. In the above example, if I usually access package/AlgorithmSolverMuchWow.java when I am in trunk/java/com/me/toplevel/, the system might recommend it to me as an autocomplete suggestion. You can add any number of other ways to enhance the experience the user has with this feature.

Data Pipelines & Distributed Systems (Algorithmic)

Suppose you want to implement a “heat map” layer on top of Google Maps. A weather heat map conceptually looks like this:



...but just the color, without the temperature labels and text.
You want to implement it as part of Google Maps, which means people could be looking at the map at extremely varied levels of zoom, from looking at the entire world on one screen, to looking at a very small region. Imagine that as part of this effort, millions of temperature sensors will be installed around the globe, at semi-regular intervals to get good coverage. The sensors will send a temperature reading with a sensor ID and temperature to a backend system every few seconds. You can look up the sensor GPS coordinates from the sensor ID because you also have a database of where each sensor was installed.

How would you process data from the sensors to create this map layer? You are responsible for the entire backend. The sensors are made by someone else and send simple REST-like calls via a mobile data connection to whatever backend you create. A couple points to start your thinking:

- The primary goal is to produce a system that is useful to users. While the closer the system is to real-time, the better, it might make it easier to only recalculate the temperature map, say, every half hour. You could consider what it would take to make it real-time, though.
- Be sure to consider how you would display the temperature at different levels of zoom. At a small enough scale, you might have pixels of the map where you don’t have any sensors. For example, what temperature reading do you display at a point if a sensor ½ mile north reads X, and another sensor ¾ of a mile east says Y, and another sensor ½ mile south-south east says Z? On the other hand, at a larger scale, the data needs to be summarized somehow.
- Consider that the amount of data coming in from this system is very large -- you will need to consider scalability.

Follow-up: Let's say we also want to create a system by which a user can select a rectangular region on the map, and learn the "average" temperature in that box. How might you implement that?