UPL Interview Prep

Riccardo Mutschlechner, Leo Rudberg, Nick Heindl

January 29, 2016

Complete the following questions as best as you can. Write out the answer (on paper) without looking anything up. A nice tip is to try to use pseudocode first to hammer out your algorithm, and then proceed to program it up in the language of your choice. Interviews often ask you to code on paper or a whiteboard, so it is essential to practice this! You should spend at most 20 minutes on each the coding/algorithmic questions.

These are some common problems we've encountered interviewing with various companies (Microsoft, Google, Dropbox, Yahoo, Riot Games, Amazon, and Uber to name a few), as well as in Cracking the Coding Interview.

The questions here cover a variety of topics, but are presented in no specific order.

If any part of any question seems vague or under-described, please ask!

1 Liftoff

Write a function that finds the first occurrence (index) of an integer k in a sorted list of integers A, with possible duplicates.

2 Abbreviation I

Write a function that takes two string arguments and returns a boolean (true/false). The function should return true if and only if the second string is a valid abbreviation of the first.

An abbreviation is defined as a substring of an original string where any missing characters are replaced a number signifying however many are missing. You can assume we will only deal with uppercase characters (case doesn't matter).

For example, ″LOCALIZATION″ is often abbreviated as ″L10N″ (disregarding case). Following the same rule, we could also have ″LOCALIZATIO1″, ″L11″, ″11N″, ″LOCALIZATION″, ″L1C1LIZATION″, and ″12″ as valid abbreviations.

For example:

```
isAbbr("LOCALIZATION", "L10N") // True
isAbbr("LOCALIZATION", "L9N") // False
isAbbr("LOCALIZATION", "L10Q") // False
```

3 Abbreviation II

Using the same rule from Abbreviation I, write a function that produces a list of all valid abbreviations of string. (Hint: there are a lot.)

For example:

```
allAbbrs("DOG") = ["DOG", "D2", ...] // order not important
```

4 The Grandfather of All Interview Questions

Given a tree of strings, write a function that determines if any grandparent has any two grandchildren that have the same value. As part of your solution, please write an accessor function for a node's grandchildren.

5 Design I

How would you implement autocomplete for a messaging app on a mobile device?

6 Design II

How would you architect an Instagram-like offering?

7 Design III

If you could change a typical programming workflow, how would you do it? In other words, how would you design a new programming environment?

8 Let's Play a Game

In the two-player game "Two Ends", n cards are laid out in a row. On each card, face up, is written a positive integer. Players take turns removing a card from either end of the row and placing the card in their pile, until all cards are removed. The score of a player is the sum of the integers of the cards in his/her pile. Give an efficient algorithm that takes A, the sequence of n positive integers, and determines the score of each player when both play optimally.

(Taken from Prof Dieter van Melkebeek's CS 577 final exam review)

9 Personality I

What project are you most proud of? Was it a success?

10 Personality II

What's the last math class you've taken? Could you apply it to software engineering? Why/why not?

11 Personality III

What’s the hardest problem you’ve ever worked on?

12 Personality IV

What would you do if you didn’t know how to do something on the job?

13 Personality V

What type of sense of humor do you have? What is funny to you?

14 Personality VI

What was the most world-changing invention in the last 100 years? 1000 years? What will the most world changing invention be in the next 100 years?

15 OOP is OP

Write a Set class that has the union, intersection, and difference operations. (Do not use a built-in Set class in your answer.)

16 Meta-testing

Write an overview of assertions you would run to test your Set implementation. You don't have to write code for this part; just the method name, the input, and the expected output of each test is fine.

17 Ask Me Later

What is a **promise** in the context of programming? Can you explain their use in five or less sentences? Can you give an example of a promise in use?

(The purpose of this question is to test your knowledge of a specific concept. Often, interviewers will ask you about topics you may never have heard of, like promises, which show up a lot front-end development as a way of handling data exchange. Please know that is OK to say ”I don’t know” in response to an interviewer’s questions.)

18 Quine Time

Write a function that gives the *n*th element in the following sequence:

The sequence begins with the element ″1″. The next item in the sequence is how that element is read: we see one ″1″, so the next element is ″11″. The element after that is ″21″ since we see two ones in the previous step. After that, we have ″1211″. The sequence continues following that pattern.

(Bonus question: do we ever see a ″4″ in a step? Why/why not?) (Double bonus: the mathematician who identified this sequence also created one of the most famous computer simulations of all time. What is his name?)

19 Pair-y Time

Given an integer k and a list of integers A, find if there exists a pair (i, j) : $A[i] + A[j] = nk$, for some $n \in Z$.