In interviews, be prepared to show that you can think and know your limits.

Taking notes from Cracking the Coding Interview.

Take my advice with a grain of salt, I've only had two technical interviews at the time of this writing and do not know everything, this is just one part of a larger scheme.

- Explain your thinking as you go. (start with big ideas and build down)
  - Plan pseudo-code or talk about the algorithm before coding.
  - Explain space efficient and time complexity (shows that you understand the concepts)
- Ask about ambiguity (or any questions)
  - Admitting that you don't know something is valuable
  - State that you would have to look something up
- Run through test runs of the code
  - look and plan test cases
  - look for fringe cases

Approaches to solving problems -

1. Examplify - start with small case and try to find general rule
2. Pattern Matching – Find similar problems and if you know the solution modify it
3. Simplify and Generalize – Think of the problem with a constrained input and modify solution for general input after solving it.
4. Base Case and Build – Solve problem for base case and build up for more complex steps
5. Data Structure Brainstorming – Run through different data structure and try each one

Tell your interviewer if you know a question (this one helped me)

Questions now

Design an algorithm and write code to remove the duplicate characters in a string without using any additional buffer. NOTE: One or two additional variables are fine. An extra copy of the array is not.

FOLLOW UP

Write the test cases for this method.

My solution –

Well, I like to start with test cases so I can know what I'm dealing with

list of test cases:

<empty string> : easy, empty string
aaa : a
abc : abc
aaccbb : acb
abcabc : abc
<arbitrarily long string> : (does it still work)

Questions I might ask –

how is the string saved? (aka, linked list, array, immutable data type, etc…)
Can I use a hash set? (Additional array of constant size)
How many unique characters are being considered?

I like to first start with a bad solution that will work –

Check every character against every previous character, if the character is found, remove it from the string. Otherwise keep it. This is a bad solution (on the order of magnitude of O(n^2) but it should work on every case).

How can I speed it up?
Let's look at the problem from a different angle –
define the problem recursively.

If the string is empty return empty string
When given a string remove the last character
Repeat this method for the substring.
Do linear search for last character on result from repeating the algorithm
if the character is not found return the string + character

Slightly better, number of characters * number of unique characters. When the number of unique characters.

I could make a O(n) if I could use a hash set. (additional array of constant size) good solution in book found on page 97 of Cracking the Coding Interview.

Sorting algorithms

Bubble Sort O(n^2) In place sort
Selection Sort O(n^2) In place sort
Merge Sort O(n log(n)) Not in place sort (But supports parallelization)
Quick Sort O(n log(n)) O(n^2) worst case In place sort
Bucket Sort O(n + m)

Brain Teasers

You are trying to cook an egg for exactly fifteen minutes, but instead of a timer, you are given two ropes which burn for exactly 1 hour each. The ropes, however, are of uneven densities - i.e., half the rope length-wise might take only two minutes to burn.

The Approach
1. What is important? Numbers usually have a meaning behind them. The fifteen minutes and two ropes were picked for a reason.
2. Simplify! You can easily time one hour (burn just one rope).
3. Now, can you time 30 minutes? That's half the time it takes to burn one rope. Can you burn the rope twice as fast? Yes! (Light the rope at both ends.)
4. You've now learned: (1) You can time 30 minutes. (2) You can burn a rope that takes X minutes in just X/2 minutes by lighting both ends.
5. Work backwards: if you had a rope of burn-length 30 minutes, that would let you time 15 minutes. Can you remove 30 minutes of burn-time from a rope?
6. You can remove 30 minutes of burn-time from Rope #2 by lighting Rope #1 at both ends and Rope #2 at one end.
7. Now that you have Rope #2 at burn-length 30 minutes, start cooking the egg and light Rope #2 at the other end. When Rope #2 burns up, your egg is done!

Next Brain Teaser (they're fun)

You have a five quart jug and a three quart jug, and an unlimited supply of water (but no measuring cups). How would you come up with exactly four quarts of water? NOTE: The jugs are oddly shaped, such that filling up exactly 'half' of the jug would be impossible

From now on – Ask class for suggestions on future steps, otherwise keep on going with this pattern:

* Another String problem, work with class
* A linked list problem, have class work on own
* Stack/queue problem, have class work on own
* Possibly Database, OO or Recursion after this
* Maybe networking and threads, but probably out of time far before this...W