

csci2200 Algorithms

Last class: Cracking a technical interview

Laura Toma
Bowdoin College

What will you remember..

- ..in one month from now?
- ..one year?
- ..5 years?

Problem solving skills.

- Mastered basic analysis and design tools
- Matured ability to think abstractly
- Appreciation of theory and its impact in practice
- Know the basic questions to ask (how to do it? why does it work? how long?)
- Can approach new problems

Algorithms questions in technical interviews

- Algorithmic questions are used in technical interviews esp. by large companies like Apple, Google, Microsoft and in general by companies who want to recruit smart people.
- Some companies might go as far as asking only conceptual questions (no programming). Usually there's a mix and most interviewers will want to see a piece of code.
- Most people will not care that you remember details (Kruskal's algorithm run in $O(E \lg V)$ time), but they want to see your thinking process. They want to see how you approach a problem.

Advice from students

- ...”Being able to solve the problem is important but what is even more important is how you approach the problem” [Anh Hoang’15, after 10+ interviews at Google, Facebook, Microsoft, Dropbox, etc]
- ..”talk through the problem..” [Andrew Daniels ’15]
-

Problems

You are presented with 9 marbles. All of the marbles look identical i.e. same shape, color, and dimensions(except for weight). However, 8 of the 9 marbles have exactly the same weight; the last marble is heavier. The only tool you have to measure weights is an old fashioned balance scale. You are only allowed to use the scale 2 times. How do you find the one marble that is not the same weight as the others?

Problems

Implement a priority queue.

Problems

Write a program to match parenthesis. Then write a program to make sure all parenthesis, curly braces and brackets are matched properly.

Problems

Given a list of license plates and a dictionary as a list, find the shortest word in the dictionary that can be made using the characters of the license plate (which may also contain numbers).

Problems

Implement a sorting algorithm on a linked list with constant memory complexity.

Problems

Implement an algorithm that returned the maximum value in a list for a "low-level library" and describe how to communicate errors to the user (i.e. by throwing an exception in java).

Then I had to extend the algorithm to return the k greatest elements in the list as fast as possible.

Problems

Given an unsorted array, rearrange the array in the form $a_1 \leq a_2 \leq a_3 \leq a_4 \dots$ in $O(n \log n)$ time. How about in linear time? (note: medium)

Problems

Given a binary tree, write a method to verify if the tree is a binary search tree in linear time in linear time and constant space. (note: between easy and medium)

Problems

Write a function to find the k th biggest number in an unsorted array in linear time (expected). (note: easy)

Problems

Given an unsorted array and a number K . Find a subset of the array whose sum is K .

(note: medium - The reason why this is medium is this is the subset sum classic problem and a lot of candidates will immediately know how to solve it)

PS: Normally the interviewer would start off asking about the 2-Sum problem, then 3-Sum and then they'd ask to generalize to find any subset. Again, it emphasizes how important the thought process is.

Problems

Given k sorted arrays, write a method to merge the into one sorted array.

(note: easy - this problem, however, is often used as an intermediate step for harder problems)

Problems

What data structure would you use to store names and phone contacts in your phone
(note: this could be a medium to hard if candidate doesn't know trie in advance)

Problems

How would you implement a data structure that can support `push()`, `pop()` and `peekMin()` in constant time?
(note: medium to hard)

Problems

Find the longest palindromic substring (contiguous) of a string.

Problems

Compare Dijkstra and A*.

Problems

Write a routine in pseudocode to reverse the bits in a word passed as input (using bitwise operations!).

Problems

You are in charge of planning a party for a company. The company has a hierarchical structure, which forms a tree rooted at the president. Each person has a supervisor, and possibly some people whom they supervise. Assume everyone is listed at the highest possible position in the tree and there are no double affiliations (everybody has only one supervisor). The company also has a (secret?) database which ranks each person with a conviviality rating (a real number, which can be negative if the person is really difficult or boring). In order to make the party fun for everybody, the president does not want both a person and his/her immediate “supervisor” to attend.

You are given a tree that describes the structure of a company. Each node also holds a name and a conviviality ranking.

Describe an algorithm to make up a guest list that maximizes the sum of the conviviality rankings of the guests. Analyze the running time of your algorithm.

Problems

Suppose you have an n -stories high building, and a bunch of eggs. An egg has a certain level l at which, if thrown from any level $\geq l$, it breaks. For example, an egg might have $l = 7$ meaning you can safely throw the egg down from levels 1 through 6, and it will not break; but if you throw the egg from a level 7 or higher, it breaks.

You are given a building and a bunch of eggs (all identical) and your goal is to find out the level l of the eggs. We can assume $n = 100$ (i.e. 100-level high building).

- Describe an approach that only breaks one egg to find out l . How many throws does it do?
- Describe an approach that minimizes the number of throws. How many eggs might it break?
- Assume now you have two eggs. Describe an approach that minimizes the number of throws. What if you had

Problems