CRACKING

ALGORITHMIC PUZZLES

TYPICAL JOB INTERVIEW QUESTION IN TOP TECH COMPANY

GIVEN AN INFINITE NUMBER OF QUARTERS (25 CENTS), DIMES (10 CENTS), NICKELS (5 CENTS) AND PENNIES (1 CENT), WRITE CODE TO CALCULATE THE NUMBER OF WAYS OF REPRESENTING N CENTS.

Interviewer

Like what?

WHY DO THEY ALL ASK THOSE CS ALGORITHM/DATA STRUCTURE QUESTIONS THESE DAYS?

- I use frameworks, and don’t need to know Computer Science things.
- I use standard library, I don’t need to know details.
- I make CRUD forms, I don’t need to know how to reverse binary tree.
- This is not related to real tasks.

REALLY WHY?

- It shows how you can recognize patterns which are standards in Computer Science and are needed in real life. Really!
- Combinatorial Search, Recursion, Divide and Conquer, as examples.
- Yes, it is about how clever you are.
- CONTRA: Risk of rejecting good candidates.

BASE CASE AND BUILD

- It is a pure math approach.
- Description: Solve the algorithm first for a base case (e.g., just one element). Then, try to solve it for elements one and two, assuming that you have the answer for element one. Then, try to solve it for elements one, two and three, assuming that you have the answer to elements one and two.
- Or solve for n, then repeat for n-1.

QUESTION

THERE ARE N STAIRS, A PERSON STANDING AT THE BOTTOM WANTS TO REACH THE TOP. THE PERSON CAN CLIMB EITHER 1 STAIR OR 2 STAIRS AT A TIME. COUNT THE NUMBER OF WAYS, THE PERSON CAN REACH THE TOP.

Interviewer

SOLUTION

- for 1 stair there is only one way to reach, just make one step.
- for 2 there are 2 ways, (1,1) or (2).
- 3 stairs is like solution for 2 and solution for 1. We have solution for it.
- And so on.
- if n =1 return 1, if n=2 return 2.
- ways(n) = ways(n-1) + ways(n-2)
- Thats all.

ACTUALLY IT WAS FIBONACCI SEQUENCE :)

Interviewer

SO THE QUESTION AGAIN...

GIVEN AN INFINITE NUMBER OF QUARTERS (25 CENTS), DIMES (10 CENTS), NICKELS (5 CENTS) AND PENNIES (1 CENT), WRITE CODE TO CALCULATE THE NUMBER OF WAYS OF REPRESENTING N CENTS.

Interviewer

SOLUTION

- We know how to solve problem for sum 1 cent.
- Check whether we have 1, if yes we return 1, cause there is only one way to have 1 cent.
- From sum subtract 25 cents, and solve it for (sum - 25), if possible.
- From sum subtract 10 cents, and solve it for (sum - 10), if possible.
- and so on…

REAL LIFE

- Why do I need them in real life?
- It is Dynamic Programming techniques, which is used in Text Processing, Optimizing Search, Combinatorial Search, A* like solution finding. Same approach, for a lot of problems. Understanding of recursive approach as well.

PATTERN MATCHING

- Consider what problems the algorithm is similar to, and figure out if you can modify the solution to develop an algorithm for this problem.

- A lot of problems possible to reduce to standard solution in Computer Science.

- Finding median, amount of distinct values in set (math set), tasks planning. All this tasks reduce to sorting problem.

- Problem solution modeling.

EXAMPLES

- Use cases in combinatorial search problems. A* algorithm in Machine Learning.

QUESTION

WRITE A METHOD TO DECIDE IF TWO STRINGS ARE ANAGRAMS OR NOT.

How about that?

SOLUTION

- Sort characters of both words.
- Compare two arrays.
- Anagrams will be equal.

QUESTION

FIND N LARGEST VALUES IN INFINITY STREAM.

How about that?

QUESTION

FIND MEDIAN IN LARGE ARRAY.

How about that?

DIVIDE AND CONQUER

- Divide problem on smaller pieces which are easy to solve.
- Merge sort example. Binary search. Distributed computing. And a lot a lot.
- Hanoi Tower again.
- Word counts. And MapReduce as well.

ACTUALLY YOU SHOULD USE A MIX OF THEM

- Directories walking, data parsing, data structures mapping and modifying, text processing, distributed systems.
- A lot of problems needs Computer Science basis for right solution.
- It is like a necessary toolbox.
- It is not just annoying interview questions.