The Wayback Machine - https://web.archive.org/web/20140321001203/http://qan…

Q&?

29 Oct 2013

I Failed a Twitter Interview

The deadline for confirming my return internship with Amazon was October 28th, but my friend Daniel convinced me that if I applied to Twitter I'd be able to finish all my interviews before then. So I went for it.

First they asked me to solve two Codility questions within a one hour time frame. The questions were interesting ("is this an anagram of a palindrome" and "count points of equilibrium in a 2d array"). I felt semi-confident about my solutions, but soon enough Judy, a lovely recruiter from Twitter, emailed me and set up a phone screening for 5:30 on Wednesday.

I don't know about you, but I get VERY nervous before interviews. I think it's mainly because I don't want the interviewer to think I'm stupid. So you can imagine, at 5:20 my desk was cleared, my notepad was labeled "Twitter Interview, Wed. Oct 23" and 2 perfectly sharpened pencils were ready for rigorous scribbling. Then came 5:30, and I started staring at my phone.

At 5:35 I googled "time in California" to make sure I calculated for the time difference right. Nope: according to Google it was 2:30 PST, 5:30 ET.

At 5:48 I emailed Judy, asking her to look into the situation. 10 minutes later I got a phone call from San Francisco. Judy apologized for the mess up and told me Justin is available to interview me right now.

Deep breath.

"Awesome, lets do it!"

Justin also apologized for the scheduling mistake, and quickly dove right into the programming question:

"Consider the following picture:"

alt text

"In this picture we have walls of different heights. This picture is represented by an array of integers, where the value at each index is the height of the wall. The picture above is represented with an array as [2,5,1,2,3,4,7,7,6]."

"Now imagine it rains. How much water is going to be accumulated in puddles between walls?"

alt text "We count volume in square blocks of 1X1. So in the picture above, everything to the left of index 1 spills out. Water to the right of index 7 also spills out. We are left with a puddle between 1 and 6 and the volume is 10."

Side note for the curious reader: I posted a gist with the correct solution to this problem at the bottom. You can keep reading without spoilers :)

The first thing I tried to do was to figure out how much water we would have at any given index. This stroke a resemblance with Calculus and integrals, so I immediately remembered that looking for local maximums could be of use. And indeed, in the picture above, the water above index 2 is bounded by the smaller of the two surrounding maximums at index 1 and 6.

I was thinking out loud: "What if we found all the local maximums, and filled in water between them. Would that work?"

"Yeah, that should work" replied Justin.

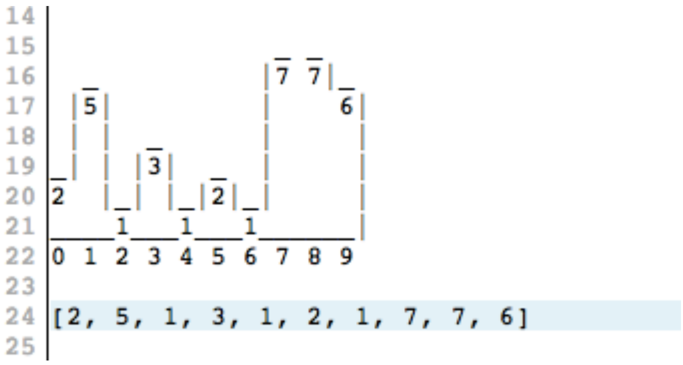
So I went ahead and coded this solution. Then Justin asked me for a bunch of test cases which I provided. All the test cases we talked about seemed to work.

"Do you have questions for me?" Justin asked. "How did I do?" "Reasonably well. Your solution does 2 passes, but there is a more interesting one that does only 1"

We then had a short chat about life at twitter.

The second I hung up I realized my solution was wrong.

Think about this input:



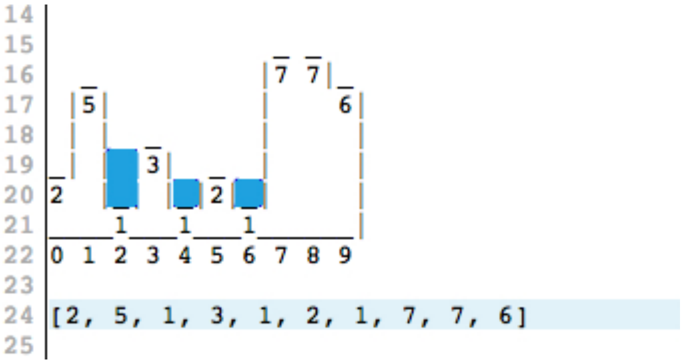
```


             |7̅ 7̅|_
 |5̅|         |    6̅|
 | |         |     |
_| |  |3̅|     |     |
2̅  |_|  |_|2̅|_|     |
____1___1___1______|
0 1 2 3 4 5 6 7 8 9

[2, 5, 1, 3, 1, 2, 1, 7, 7, 6]

```

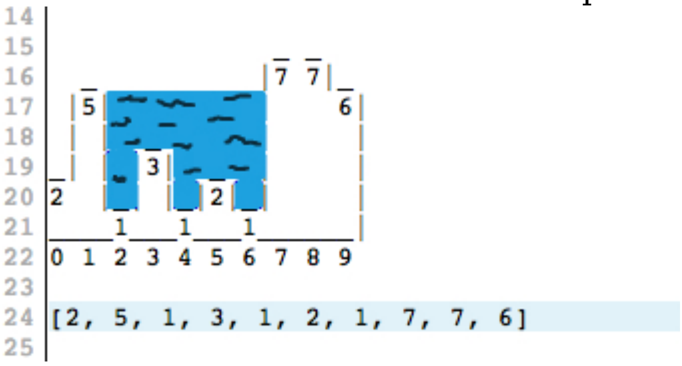
My solution solved between the local maximums and looked like this:



```
              _ _
  _          |7 7|_
 |5|         |    6|
 | |  _      |     |
_| |  |3|  _  |     |
2  |_|  |_|2|_|     |
____1___1___1______|
0 1 2 3 4 5 6 7 8 9

[2, 5, 1, 3, 1, 2, 1, 7, 7, 6]
```

But the result should have been one puddle between the two taller towers:



```
              _ _
  _          |7 7|_
 |5|         |    6|
 | |  _      |     |
_| |  |3|  _  |     |
2  |_|  |_|2|_|     |
____1___1___1______|
0 1 2 3 4 5 6 7 8 9

[2, 5, 1, 3, 1, 2, 1, 7, 7, 6]
```

The next day I showed this question to my TA, who's a PhD student in theoretical computer science. After 40 minutes he was also stuck with nothing.

Today I woke up with smelly breath and a eureka. The solution is beautiful and simple.

Now I ask myself: what have I learned from this? Realistically - not much. I am upset that the interviewer didn't ask me the right questions to guide me towards the right train of thought. I don't know why Justin told me "this should work," when my solution in fact didn't . I know that this should have come up in the test cases he asked for, but since I missed the flaw when coming up with the algorithm, I didn't think of testing for it.

I signed the contract with Amazon for next summer and I'm really excited about that! At the same time I can't help but ask 'what if?'

Here's the gist for the solution.

The logic is as follows:

If we traverse the list from left to right, the amount of water at each index is *at most* the largest value we have discovered so far. That means that if we knew for a fact that there is something larger or equal to it somewhere on the right, we would know exactly how much water we can hold without spilling. Same goes for traversing in the opposite direction: if we know we have found something larger on the left than the largest thing on the right, we can safely fill up water.

With this in mind, one solution would be to first find the absolute maximum value, traverse from the left to the maximum, and then traverse from the right to the maximum. This solution does 2 passes: one to find the maximum, and the other is split into two subtraversals.