

Topics:

- **Where to find jobs to apply to**
- What to put on your resume
- Code screening
- Cover Letter
- Behavioral interview questions (could potentially combine with HR interview questions)
- HR interview questions (“walk us through your background”, “would you take that [summer internship→full-time offer] position if offered”)
- ML+stats concepts (for data science related positions)
- Brain teasers (finance positions)
- Take Home Assessments
- Salary Negotiation

**where to find jobs to apply to:**

**Look on LinkedIn**

**Check out Glassdoor and**

**Indeed (possibly Zip Recruiter) for positions**

**Google New Grad Positions Github**

**Todo: Write a first draft of a resume, and find articles on resumes and cover letters.**

### **Notes from discussion on Resumes**

- Putting your address, you could put your city and state instead of the full address
- Email accounts: eventually the usc account will become deactivated so it's better to use a personal account that looks professional
- Skills: Do you list your proficiency?
- You can find resume templates in the templates section of overleaf
- If it's a coding related job link your github to the resume! I had an interview today where the guy was saying how he reads so many resumes and no one links the github and he was like "how do I know they can code otherwise??" So it can really help you get past the resume screen.

### **Notes from discussion on HR Interview**

- Share your background and have an example or two or a story for why you want to go into that industry.
- Be able to answer "What do you know about our company" (just a surface-level answer will do)
- What is your expected compensation? (Respond by asking if there's a range for the position, if so then say that range sounds good otherwise give a wide range and say it's open to discussion based on the details of the job. Alternatively you could research what the range is beforehand and just give that)
  - For salary info, go to levels.fyi, salarylookup.net (h1b reporting), and glassdoor reports
- If it's an app, have it downloaded on your phone and be prepared to talk about a feature you'd implement and how you'd go about doing it.
- Compare and contrast the app with their competitors.
-

## Sample Coding Questions

You are given a string  $S$  of length  $N$  which encodes a non-negative number  $V$  in a binary form. Two types of operations may be performed on it to modify its value:

- if  $V$  is odd, subtract 1 from it;
- if  $V$  is even, divide it by 2.

These operations are performed until the value of  $V$  becomes 0.

For example, if string  $S = "011100"$ , its value  $V$  initially is 28. The value of  $V$  would change as follows:

- $V = 28$ , which is even: divide by 2 to obtain 14;
- $V = 14$ , which is even: divide by 2 to obtain 7;
- $V = 7$ , which is odd: subtract 1 to obtain 6;
- $V = 6$ , which is even: divide by 2 to obtain 3;
- $V = 3$ , which is odd: subtract 1 to obtain 2;
- $V = 2$ , which is even: divide by 2 to obtain 1;
- $V = 1$ , which is odd: subtract 1 to obtain 0.

Seven operations were required to reduce the value of  $V$  to 0.

Write a function:

```
def solution(S)
```

Write a function `solution` that, given an integer  $N$ , returns the maximum possible value obtainable by deleting one '5' digit from the decimal representation of  $N$ . It is guaranteed that  $N$  will contain at least one '5' digit.

Examples:

1. Given  $N = 15958$ , the function should return 1958.
2. Given  $N = -5859$ , the function should return -589.
3. Given  $N = -5000$ , the function should return 0. After deleting the '5', the only digits in the number are zeroes, so its value is 0.

Assume that:

- $N$  is an integer within the range  $[-999,995..999,995]$ ;
- $N$  contains at least one '5' digit in its decimal representation;
- $N$  consists of at least two digits in its decimal representation.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Task 1

Python

English

Files

Find the bug(s) and modify one line of code in the incorrect implementation of a function `solution` that is supposed to return the smallest element of the given non-empty array  $A$  which contains at most 1000 integers within range  $[-1000..1000]$ .

Notice that for the example test case  $A = [-1, 1, -2, 2]$  the attached code is already returning the correct answer (-2).

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

task1

solution...

test-input.txt

solution.py x

```
1 1 def solution(A):
2 2     ans = 0
3 3     for i in range(1, len(A)):
4 4         if A[i] < ans:
5 5             ans = A[i]
6 6     return ans
7 7
```

Leetcode.com to familiarize yourself with the terminal. This quiz was on codility

List of top 75 leetcode questions Blind

<https://github.com/DopplerHQ/awesome-interview-questions#python>

**To practice SQL: hackerrank.com has a decent database of SQL problems, leetcode has sql problems but you need to filter by the category “database”.**

**To learn SQL, Kaggle has two really good microcourses (Kaggle has really good microcourses for getting started with coding and data science in general)**

**General Tips:**

- **Solve it mathematically if the fastest algorithm you can think of doesn't work. (so if you have a solution that is linear in time complexity then realistically the only thing faster would be a direct formula)**
- **Sometimes the test cases that the company gives you are not good enough. So it's a good idea to think of edge cases. The company might have a larger list of test cases that they use to evaluate your code.**
- **On the code screen, if it's online, then debugging is very simple because you can just have the program output variables that you think are acting strange. This is not possible on a whiteboard interview so when preparing it's best to prepare as if it were a whiteboard interview**

Question: “What is an object oriented programming language”

Ans: “An oop is a programming language where you have classes that output objects with attributes, for example a class where the objects are students and their attributes are name and grade. On the other hand a functional programming language is where all of the actions are done by functions, so for example instead of using for loops, and while loops, you would use a recursive function to achieve the same thing”

Question: How do you handle data where there's lots of features?

Ans: So in my previous internship I had many features to consider so I focused on the business aspect of it by speaking with members of different teams to figure out which features would be most relevant. In this case since there's a much larger number of features, I would still first understand what is important from a business POV but then I would try to add some other model selection techniques, such as “forward selection” and “backwards selection” in a linear regression for example.

To add, there are many more techniques. For example you can look at which features are most highly correlated and use that to eliminate some of them. You could also do a Principal Component Analysis(PCA)\* (May not work because the correlation matrix would be large). Decision tree classifiers can handle lots of features well

**Pytorch:**

**[youtu.be/BzcBsTou0C0](https://youtu.be/BzcBsTou0C0)**

**Fast.ai does their stuff in pytorch.**

**Tensorflow:**

**Machine learning with Scikit-Learn and Tensorflow**

**There's also a tensorflow developer certification that google offers (I think its free but I could be wrong)**

**Overall I suggest doing a project to learn the skills**

**Math550**

**<https://www.statlearning.com/>**

**Elements of Statistical Learning (a good reference book but focus more on projects and coding)**

**Look up Data Science Cheatsheet 2.0**

**What classifies as a worthy project?**

**Examples: John's project was looking at the abstracts on arxiv, and making a classifier based on that. So it can identify what subject the paper belongs to(within math). 1. Constructing the data himself, he had to learn how to use an API and make calls, so it shows that he can really deal with messy data. 2.It's a nontrivial deep learning exercise (classifying different subjects). 3. He made a web application that allows you to copy paste the abstract and then it will identify the subject classification.**

**Other examples: you can also look at my github which I linked on the resume. (apoorvashah101) (there was CharityML , another was a flower image classifier)(my favorite projects were Textual Data Augmentation, Clinical Notes)**

**A common project will be doing something with the Titanic Dataset**

**Another example: 1992 LA Riots Project (you can see this on my kaggle apoorvashah101)**

**Salary Negotiation:**

- **Ask for a little more than what you actually want**
- **<https://haseebq.com/my-ten-rules-for-negotiating-a-job-offer/>**
- **<https://haseebq.com/how-not-to-bomb-your-offer-negotiation/>**
- **Negotiating your Salary: How to make \$1000 a minute by Jack Chapman**
- **pon.harvard.edu/blog**
- **Know what your compensation should be for your background.**
- **Calculate how long you can spend on the job hunt depending on your situation.**
- **Talk to actual decision makers**
- **Time it so that you have multiple offers, because it offers you much much much more leverage.**
- **Reach out to alumni/alumni events**
- **Alumni database (especially non-academic industry jobs)**

### Sample Problems from interviews ( Non-Coding):

1. You are at a bus stop. If you have been waiting for at least  $x$  minutes, your expected wait time is another  $x$  minutes. Find the distribution of your waiting time.
  - a.  $E[X|X>x] = 2x$
2. There is a bar with 25 seats equally spaced on a line. The first person is assigned a seat by the bartender. The next person chooses a seat that maximizes the minimum distance between anyone else. People will not sit next to each other. Where should the first person be placed to maximize the number of seats filled.
3. 20 equally spaced points on a circle. You make two chords by connecting the two points(4 distinct points in total). What's the probability that the chords intersect?
4. Linear OLS Regression. You are trying to fit a model  $Y = bX$  and instead you copy your  $X$  column again to try and fit the model  $Y = b_1X_1 + b_2X_2$ . ( $X_1$  and  $X_2$  are two copies of the same column vector). i) how do  $b_1$  and  $b_2$  relate to  $b$ ? ii) Now suppose it was a lasso regression? What about ridge regression?
5. (Essentially St.Petersburg paradox) What would you do in a game with infinite expectation. (scenario: you are flipping a coin and you stop flipping once you get tails. Your payout is the number of heads. Now what if your payout is  $2^{(\text{number of heads})}$  how much would you pay to play this game )
6. There's a horse race with 4 horses A, B, C,D. The odds are 1:2, 1:3, 1:4 and 1:5. You have \$100 to bet. What would you do? You pay 100 and if A wins then they pay you \$300 otherwise, they pay you 0. What do you do? And why does it work that way?
  - a. Idea: bet 100/3 on horse A. 100/4 on horse B, 100/5 on horse C and then 100/6 on horse D. This only adds up to 95.
  - b. Federer vs Nadal 40/60 and another website says it's 50/50
7. Two people toss a fair coin one after another; whoever flips tails after heads wins. What is the probability that the first person tossing wins? Ans: 4/9