PDP ONLINE

DSA - Algorithms Introduction

Learning Paths

Learning Path-1

1- Algorithms

2- Data Structures

3- Web, Mobile

4- System Designs

Learning Path-2

1- Web, Mobile

2- Algorithms

3- Data Structures

4- System Designs

Course Planning

Algorithms	Data Structures	Algorithmic Approaches	Interview Practices
1.Introduction	1.Asymptotic Analysis	1.Search Algorithms	1.In-place Reversal
2.Number 1	2.Dynamic Array	2.Sort Algorithms	2.Two Heaps
3.Number 2	3.LinkedList	3.Dac Algorithms	3.Subsets
4.String 1	4.Stack	4.Recursion	4.Modified BS
5.String 2	5.Queue	5.Sliding Window	5.Bitwise XOR
6.Array 1	6.HashTable	6.Two Pointers	6.Top 'K' Elements
7.Array 2	7.Tree	7.Fast & Slow	7.K-way Merge
8.Matrix	8.Trie	8.Cyclic Sort	8.Knapsack Problem
9.DP 1	9.Directed Graph	9.Breadth First Search	9.Topological Sort
10.DP 2	10.Undirected Graph	10.Depth First Search	10.Mock Interview

https://kurbanovxurshid.medium.com/data-structures-and-algorithms-course-plan-1ab912ec1e17

Mock Interview

It's your turn to be the Interviewer. The question is available in the left pane below the video

Language: Python ▼ | Swap Roles | End Interview

```python
# logs storage
M = [[0 for x in range(n)] for x in range(n)]


# Reset logs
def resetLogs():
    for i in range(0,n):
        for x in range(0,n):
            M[i][j] = 0


# Log a call between 2 users
def logCall(source, destination):
    M[source][destination] = M[source][destination] + 1


# get the amount of calls btween 2 users
def getCallCount(source, destination):
    print (M[source][destination])
```

The "Call Logger System Design" Problem

You are asked to plan a call log system that logs info about calls between `n` users.
The system should support the following interface:

- Log a call from user `Ux` to user `Uy`.
- Figure how many calls, if at all a given user `Ux` have made to a given user `Uy`
- Provide the list of all users that a given user `Ux` has called to
- Reset all logs and memory of the system

Design the system, choose appropriate data structures and implement the code to support the interface above. Analyze the runtime and space complexity of each interface method you implement.

Solution

The data could be represented by using a 2D matrix of integers of size `n X n`.
The integer value on a matrix cell `M[x,y]` will represent the number of calls that user `Ux` has made to user `Uy`.

Reset action will be done by writing zeros into all cells. This action will also be done when the system is first initiated:

```
function resetLogs():
    for i from 0 to n-1:
```

Tips

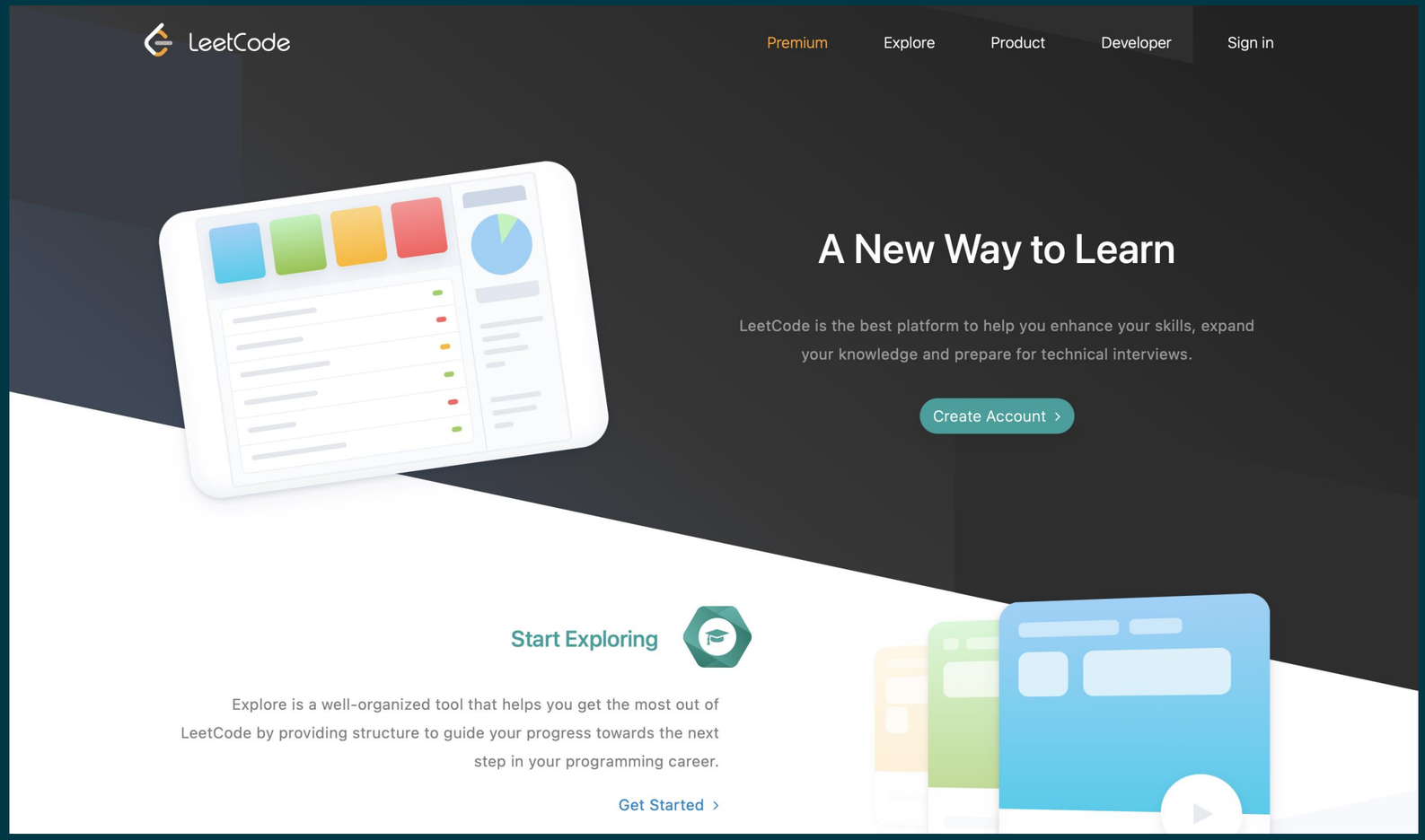
- Encourage your peer to optimize for of runtime and space complexity combined
- If your peer is stuck, give an example of few calls and ask what would the interface actions do on this data set. Then ask how can you represent this data
- If you peer is still stuck, ask them to choose the most basic representation (even an array for every user on the system) and start improving it

Asked by Companies

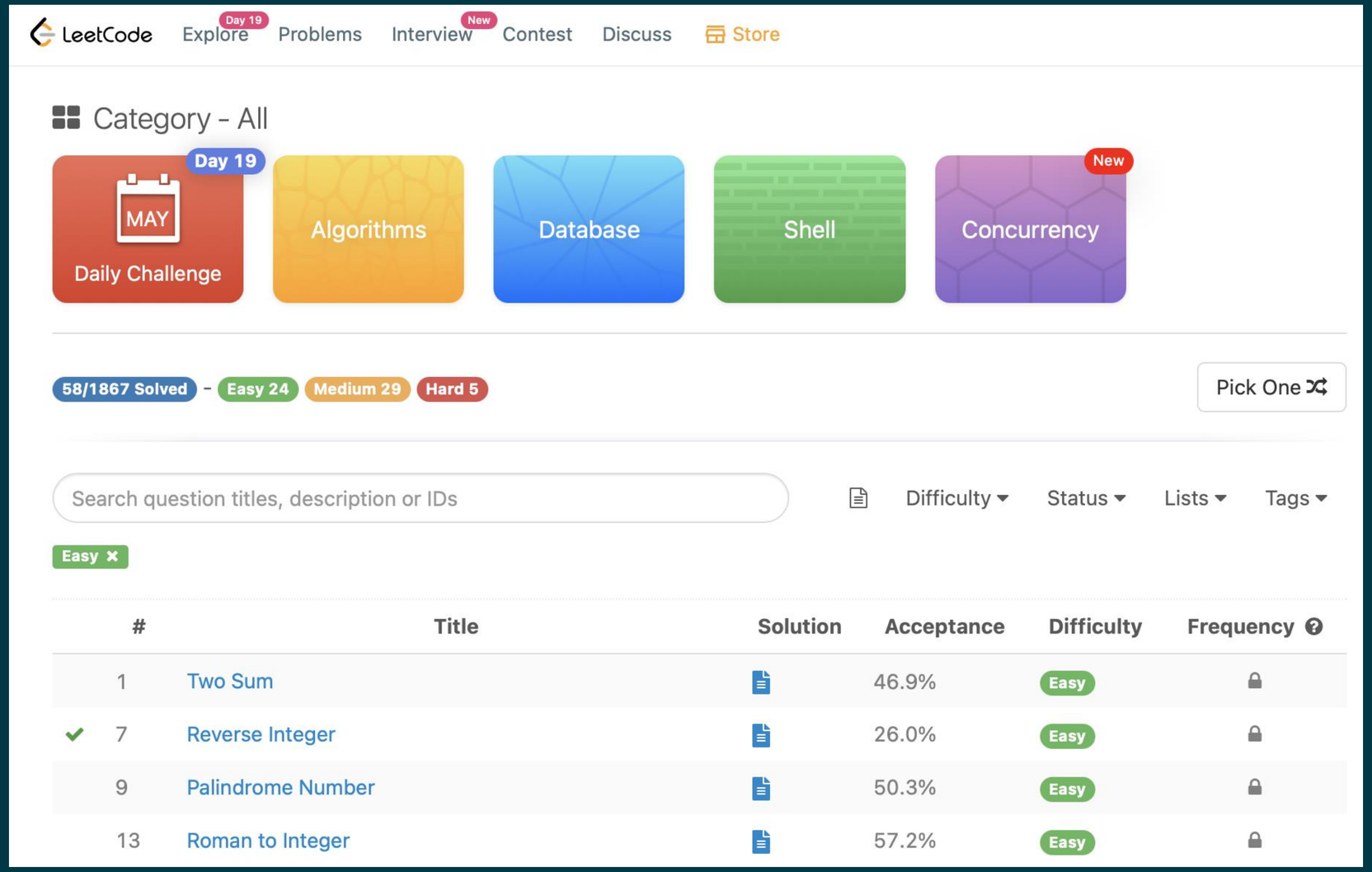



5 Websites for DSA

LeetCode Platform



LeetCode Problems



Coding Challenge

7. Reverse Integer

Easy 4756 7250 Add to List Share

Given a signed 32-bit integer `x`, return `x` *with its digits reversed*. If reversing `x` causes the value to go outside the signed 32-bit integer range `[-2^31, 2^31 - 1]`, then return `0`.

Assume the environment does not allow you to store 64-bit integers (signed or unsigned).

Example 1:

```
Input: x = 123
Output: 321
```

Example 2:

```
Input: x = -123
Output: -321
```

Example 3:

```
Input: x = 120
Output: 21
```

https://leetcode.com/problems/reverse-integer/

Test Cases


class Solution {
public int reverse(int x) {
long res = 0;
while(x != 0){
res = res*10 + x%10;
x = x / 10;
}
if(res < Integer.MIN_VALUE || res > Integer.MAX_VALUE)
return 0;
return (int)res;
}
}
Your previous code was restored from your local storage. Reset to default
Testcase
Run Code Result
Debugger
Accepted
Runtime: 0 ms
Your input
123
Output
321
Diff
Expected
321

Wrong Answer

Wrong Answer Details ›

Input `1534236469`

Output `1056389759`

Expected `0`

Time Submitted	Status	Runtime	Memory	Language
05/19/2021 22:47	Wrong Answer	N/A	N/A	java
05/19/2021 22:43	Accepted	1 ms	36.3 MB	java
05/19/2021 22:42	Accepted	1 ms	35.8 MB	java
05/19/2021 22:42	Wrong Answer	N/A	N/A	java
05/19/2021 22:42	Wrong Answer	N/A	N/A	java

Success Answer

Success Details ›

Runtime: 1 ms, faster than 100.00% of Java online submissions for Reverse Integer.

Memory Usage: 36.1 MB, less than 57.67% of Java online submissions for Reverse Integer.

Next challenges:

String to Integer (atoi)

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
05/19/2021 22:47	Accepted	1 ms	36.1 MB	java
05/19/2021 22:47	Wrong Answer	N/A	N/A	java
05/19/2021 22:43	Accepted	1 ms	36.3 MB	java
05/19/2021 22:42	Accepted	1 ms	35.8 MB	java

Homework Rating

1- Accepted

2- Faster than 50%

Task 1 – Fibonacci Number

509. Fibonacci Number

Easy 1363 225 Add to List Share

The **Fibonacci numbers**, commonly denoted `F(n)` form a sequence, called the **Fibonacci sequence**, such that each number is the sum of the two preceding ones, starting from `0` and `1`. That is,

```
F(0) = 0, F(1) = 1
F(n) = F(n - 1) + F(n - 2), for n > 1.
```

Given `n`, calculate `F(n)`.

Example 1:

```
Input: n = 2
Output: 1
Explanation: F(2) = F(1) + F(0) = 1 + 0 = 1.
```

Example 2:

```
Input: n = 3
Output: 2
Explanation: F(3) = F(2) + F(1) = 1 + 1 = 2.
```

https://leetcode.com/problems/fibonacci-number/

Task 2 – Count Primes

204. Count Primes

Easy 3208 801 Add to List Share

Count the number of prime numbers less than a non-negative number, `n`.

Example 1:

```
Input: n = 10
Output: 4
Explanation: There are 4 prime numbers less than 10, they are 2, 3, 5, 7.
```

Example 2:

```
Input: n = 0
Output: 0
```

Example 3:

```
Input: n = 1
Output: 0
```

https://leetcode.com/problems/count-primes/

Task 3 – Roman to Integer

13. Roman to Integer

Easy 626 62 Add to List Share

Roman numerals are represented by seven different symbols: `I`, `V`, `X`, `L`, `C`, `D` and `M`.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, `2` is written as `II` in Roman numeral, just two one's added together. `12` is written as `XII`, which is simply `X + II`. The number `27` is written as `XXVII`, which is `XX + V + II`.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not `IIII`. Instead, the number four is written as `IV`. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as `IX`. There are six instances where subtraction is used:

- `I` can be placed before `V` (5) and `X` (10) to make 4 and 9.
- `X` can be placed before `L` (50) and `C` (100) to make 40 and 90.
- `C` can be placed before `D` (500) and `M` (1000) to make 400 and 900.

Given a roman numeral, convert it to an integer.

https://leetcode.com/problems/roman-to-integer/