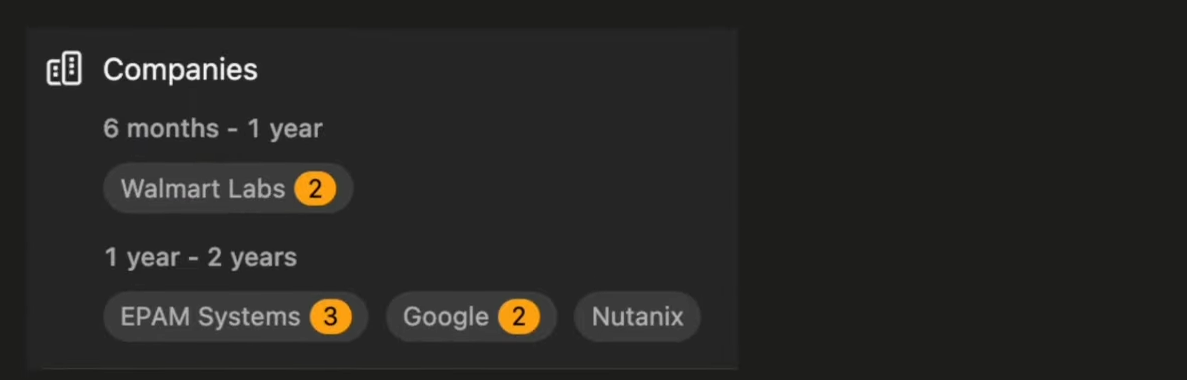
**Leetcode Daily Problems:**

**21st June 2024**

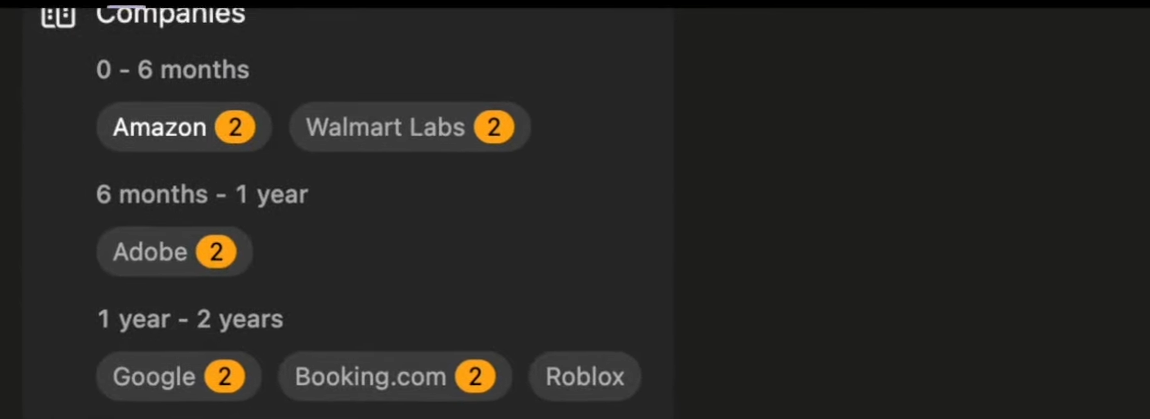
**Q) Grumpy Bookstore Owner:**



* This problem is solved using sliding window technique.
* In this problem there is a grumpy bookstore owner and there are customers coming in bookstore every minute.
* If owner is grumpy at that minute they are not satisfied otherwise satisfied.
* The owner however can stay non grumpy for minutes duration only once if he wants.
* What is the maximum number of satisfied customers.
* So to solve this problem we fill calculate all those customers which came when he wasn’t grumpy.
* Then we will create a sliding window of length minutes and move it taking the sum of customers which came when he was grumpy.
* We will calculate the maximum sum available for this window.
* While moving the window when window is ahead of the 0th minute we have to add the next element and delete the value of last element if the owner was grumpy then.
* This way we maximize the value from the window and at that duration he will use his ability to stay non grumpy.

**22nd June 2024**

**Q) 1248 Count number of Nice Subarrays: (Imp)**



* This is the same question as 560 subarray sum equal to k and 930 Binary Subarrays with sum.
* In this problem we need to find the count of nice subarrays in the given array.
* A continuous subarray is called nice if there are k odd numbers on it.
* We only care about odd and even in this question.
* We need k odd numbers present in the subarray for it to be nice.
* So we convert the whole array to binary 0’s and 1’s as then we can easily solve this question as it becomes question 930 of leetcode.
* So we first convert it to binary and then we apply the concept of finding subarray sum equal to k.
* Since all odd numbers are 1 and we need k odd numbers so we need k 1’s.
* So we calculate the presume and put it in unordered map.
* We store (0,1) already in the map.
* Then we check the presum value. Then we calculate the remainder value and check if it exists in the map and if it does then how many times that sum was calculated and then we add in count the number of times it appeared and this gives the solution.
* This solution works for 0’s positives as well as negatives and since in this question we only have positives and zeroes this works perfectly fine.
* Another solution can be working with sliding window. We keep two variables left and right and move right to increase value of presum and move left to decrease value of presum.
* But in this question if we calculate it like this we will skip a few possible solutions as if we expand first we miss solutions which could have been achieved if we compress first since when we increase right or left value the possible presum can change and can remain same. So that condition of remaining same can cause problems.
* So instead we calculate all possible nice subarrays with less than equal to k odd numbers and then we calculate all possible nice subarrays with less than equal to k-1 odd numbers and then subtract these both to get exact k odd numbers.
* **This is a very important question.**

**27th June 2024**

**1791 Find center of star graph**

* This is a problem of graph in which we are given a star graph and we need to return the node of the center of the star what is its value.
* Since in a star the center is connected to all the values so in the edges there will be a node that will appear everytime and that will be our center node.
* So we store the value of first edges in temp 1 and temp2 and then we check the next node if it contains either temp1 or temp2 and it must contain any one of them as all edges will contain a common node.
* Return that common node as the center.

**10th July 2024**

[**1598. Crawler Log Folder**](https://leetcode.com/problems/crawler-log-folder/)

* In this problem we have to return how many layers are we inside the folder and we are given 3 operations.
* To solve this problem we need to run a loop on the vector strings and every time we name a folder we increase count by 1 and we decrease count by 1 when we go back a folder.
* But make sure you don’t go below 0.
* In the end return the count value.