DSA SERIES

- Learn Coding

Topic to be Covered today

Prefix Sum Problems

LETS START TODAY'S LECTURE

Prefix Sum

A **prefix sum** is an array (or sometimes a formula) that stores the cumulative sum of a given array up to each index.

- **Each element of the prefix sum array tells you the sum of all elements before (and including) that index.**

Given an array: arr = [2, 4, 1, 3, 6]

We want to compute the sum from index i to j (inclusive) multiple times quickly.

```
vector<int> prefix(arr.size());
prefix[0] = arr[0];

for (int i = 1; i < arr.size(); i++) {
    prefix[i] = prefix[i - 1] + arr[i];
}
```

Problem : 2483

Minimum Penalty for a Shop

Code

```
class Solution {
public:
    int bestClosingTime(string customers) {
        int n = customers.size();

        vector<int> prefix(n+1,0);  // No of N in 0 to j-1 index
        vector<int> suffix(n+1,0); // No. of Y in j to n index

        for(int i = 0;i<n;i++){
            prefix[i+1] = prefix[i]+(customers[i]=='N'?1:0);
        }

        for(int i = n-1;i>=0;i--){
            suffix[i]=suffix[i+1]+(customers[i]=='Y'?1:0);
        }
```

```
            int minPenalty = INT_MAX;
            int bestHour = 0;

            for(int  i =0;i<=n;i++){
                int penalty = prefix[i]+suffix[i];

                if(penalty < minPenalty){
                    minPenalty = penalty;
                    bestHour = i;
                }
            }

            return bestHour;
        }
    };
```

Problem : 2100

Find Good Days to Rob the bank

```
class Solution {
public:
    vector<int> goodDaysToRobBank(vector<int>& security, int time) {
        int n = security.size();

        vector<int> prefix(n, 0);
        vector<int> suffix(n, 0);

        for (int i = 1; i < n; i++) {
            if (security[i] <= security[i - 1]) {
                prefix[i] = prefix[i - 1] + 1;
            } else {
                prefix[i] = 0;
            }
        }

        for (int i = n - 2; i >= 0; i--) {
            if (security[i] <= security[i + 1]) {
                suffix[i] = suffix[i+1] + 1;
            } else {
                suffix[i] = 0;
            }
        }
```

```
        vector<int> result;

        for(int i = time ;i<n-time;i++){
            if(prefix[i]>=time && suffix[i]>=time){
                result.push_back(i);
            }
        }

        return result;
    }
};
```

Problem : 1685

Sum of Absolute Differences in a Sorted Array

```
class Solution {
public:
    vector<int> getSumAbsoluteDifferences(vector<int>& nums) {
        int n= nums.size();

        vector<int> ans;
        vector<int> left(n,0);
        vector<int> right(n,0);

        for(int i = 1;i<n;i++){
            left[i]=left[i-1]+nums[i-1];
        }

        for(int i = n-2;i>=0;i--){
            right[i]=right[i+1]+nums[i+1];
        }

        for(int i = 0;i<n;i++){
            int sum = abs(nums[i]*i - left[i]);
            sum+= abs(right[i]-nums[i]*(n-i-1));

            ans.push_back(sum);
        }

        return ans;
    }
};
```

Learn coding

THANK YOU