



# Prefix

## Sum

# What is Prefix Sum?

arr =      0    1    2    3    4    5    6  
         { 5, 1, 8, 2, 4, 3, 2 }

pre =    { 5, 6, 14, 16, 20, 23, 25 }

$$\text{pre}[i] = \text{arr}[i] + \text{pre}[i-1]$$

# Ques:

## Q1 : Running sum of 1D Array

$arr = \{3, 4, 6, 16, 17\}$   $\rightarrow$  Extra Space  
 $\downarrow$   
 $O(1)$

$arr[i] += arr[i-1]$

[Leetcode 1480]

# Ques:

## Q2 : Range Sum Query <sup>Im</sup> - Mutable

arr = { 0 1 2 3 4 5 6  
5, 1, 8, 2, 4, 3, 2 } length = n

left, right → 2 indices  
1 1  
2 5

'm' pairs of (l, r)

If we calculate the sum  
from left to right using  
for loop, then T.C. =  $O(n)$   
for one query

$$T.C. = O(m * n)$$

[Leetcode 303]

# Ques:

## Q2 : Range Sum Query - Mutable

arr =  $\begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \{ & 5, & 1, & 8, & 2, & 4, & 3, & 2 \} \end{matrix}$

pre =  $\{ 5, 6, 14, 16, 20, 23, 25 \}$

(2, 5) sum

↓

$(0, 5) - (0, 1)$

$$\text{sum}(\text{left}, \text{right}) = \text{sum}(0, \text{right}) - \text{sum}(0, \text{left} - 1)$$

[Leetcode 303]

# Homework:

Q: Find Pivot Index

arr = 

0	1	2	3	4	5	
	1	7	3	6	5	6

ans = 3

10, 99, 1, 2, 3, 4  
↓  
pivot

arr = {10, 1, 2, 3, -6}

arr = {6, -3, -1, -2, 10}

[Leetcode 724]

# Homework:

## Q : Find Pivot Index

```
public int pivotIndex(int[] nums) {  
    int n = nums.length;  
    for(int i=0;i<n;i++){  
        int leftSum = 0, rightSum = 0;  
        for(int j=0;j<i;j++){  
            leftSum += nums[j];  
        }  
        for(int j=i+1;j<n;j++){  
            rightSum += nums[j];  
        }  
        if(leftSum==rightSum) return i;  
    }  
    return -1;  
}
```

Time Complexity :  $O(n^2)$

Extra Space :  $O(1)$

[Leetcode 724]

# Homework:

**Q : Find Pivot Index**

arr =       $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \\ \{ 1, & 7, & 3, & 6, & 5, & 6 \} \end{matrix}$

pre =       $\{ 1, 8, 11, 17, 22, 28 \}$

$$\rightarrow \text{leftsum} = \text{pre}[i-1]$$

$$\rightarrow \text{rightsum} = \text{sum}(i+1, n-1) = \text{sum}(0, n-1) - \text{sum}(0, i) \\ \text{pre}[n-1] - \text{pre}[i]$$

[Leetcode 724]



# Homework:

## Q : Find Pivot Index

```
public int pivotIndex(int[] nums) {  
    int n = nums.length;  
    // make nums prefix sum of nums  
    for(int i=1;i<n;i++){  
        nums[i] += nums[i-1];  
    }  
    for(int i=0;i<n;i++){  
        int leftSum = 0;  
        if(i>0) leftSum = nums[i-1];  
        int rightSum = nums[n-1] - nums[i];  
        if(leftSum==rightSum) return i;  
    }  
    return -1;  
}
```

Time Complexity :  $O(n)$

Extra Space :  $O(1)$

[Leetcode 724]

# Homework:

**Q : Find the Score of all Prefixes of an Array**

$$\text{arr} = \{ 7, 2, 10, 5, 3 \}$$

$$\text{arr2} = \{ 14, 9, 20, 15, 13 \}$$

$$\text{arr3} = \{ 14, 23, 43, 58, 71 \}$$

$$\text{arr2}[i] = \text{arr}[i] + \max(0, i)$$

**[Leetcode 2640]**

# Ques:

**Q3:** Product of array except self [Pure Interview]

$$\text{arr} = \{ 7, 2, 4, 3 \}$$

$$\text{product} = 168$$

$$\text{ans} = \{ 24, 84, 42, 56 \}$$

$$\text{ans}[i] = \text{pro} / \text{arr}[i]$$

[Leetcode 238]

# Ques:

**Q3:** Product of array except self

$arr = \{7, 2, 4, 3\}$

$sufsum = \{16, 9, 7, 3\}$

$suf[i] = arr[i] + suf[i+1]$

$presum = \{7, 9, 13, 16\}$

Prefix Sum,

Suffix Sum

[Leetcode 238]

**Ques:**

$$pre[i] = arr[i] * pre[i-1]$$



**Q3: Product of array except self**

$$arr = \{ 24, 56, 84, 42 \}$$

$$arr = \{ 7, 3, 2, 4 \}$$

$$sufpro = \{ 168, 24, 8, 4 \}$$

$$prepro = \{ 7, 21, 42, 168 \}$$

$$arr = \{ 7, 3, 2, 4 \}$$

$$pre = \{ 1, 7, 21, 42 \}$$

$$suf = \{ 24, 8, 4, 1 \}$$

$$pre[i] = pre[i-1] * arr[i-1]$$

$$suf[i] = suf[i+1] * arr[i+1]$$

**[Leetcode 238]**

# Ques:

## Q3: Product of array except self

```
public int[] productExceptSelf(int[] arr) {  
    int n = arr.length;  
    int[] pre = new int[n];  
    pre[0] = 1;  
    for(int i=1;i<n;i++){  
        pre[i] = arr[i-1] * pre[i-1];  
    }  
    int[] suf = new int[n];  
    suf[n-1] = 1;  
    for(int i=n-2;i>=0;i--){  
        suf[i] = arr[i+1] * suf[i+1];  
    }  
    int[] ans = new int[n];  
    for(int i=0;i<n;i++){  
        ans[i] = pre[i] * suf[i];  
    }  
    return ans;  
}
```

$$T.C. = O(n)$$

$$S.C. = O(n)$$

[Leetcode 238]

# Ques:

## Q3: Product of array except self

$$\text{arr} = \{ 7, 3, 2, 4 \}$$

$$\text{pre} = \{ 1, 7, 21, 42 \}$$

$$\text{suf} = \{ 24, 8, 4, 1 \}$$

$$\text{ans} = \{ 24, 56, 84, 42 \}$$

$$\text{arr} = \{ 7, 3, 2, 4 \}$$

$$\text{pre} = \{ 1, 7, 21, 42 \}$$

$$\text{suf} = 1;$$

$$\text{pre}[i] *= (\text{suf} * \text{arr}[i+1])$$

# Ques:

## Q3: Product of array except self

```
public int[] productExceptSelf(int[] arr) {  
    int n = arr.length;  
    int[] pre = new int[n];  
    pre[0] = 1;  
    for(int i=1; i<n; i++){  
        pre[i] = arr[i-1] * pre[i-1];  
    }  
    int suf = 1;  
    for(int i=n-2; i>=0; i--){  
        suf *= arr[i+1];  
        pre[i] *= (suf);  
    }  
    return pre;  
}
```

Best Method

↓

T.C. =  $O(n)$

S.S. =  $O(n)$

[Leetcode 238]



**Ques:** Sort, Prefix Sum, B.S.

**Q4 :** Longest subsequence with limited sum  
↑ subset

arr = {4, 5, 2, 1, 2, 7}

queries = {5, 10, 11}

ans = {3, 4, 4}

4, 5, 2

7, 4

7, 2, 2

5, 2, 1, 2

1

2

1, 2

2, 2

1, 2, 2 → length = 3

4

5

4, 1

[Leetcode 2389]

$arr = \{4, 5, 2, 1, 2, 7\} \xrightarrow{\text{sort}} arr = \{1, 2, 2, 4, 5, 7\}$

$$\text{ans} = \{3, \underset{4}{2}, \underset{4}{2}\}$$

↓ pre

arr = { 1, 3, 5, 9, 14, 21 }

if(arr[mid] > que[i]) hi = mid - 1

```
else {
    ans[i] = max(ans[i], mid + 1);
    lo = mid + 1;
}
```

[Leetcode 2

## [Leetcode 2389]

# Ques:

## Q4 : Longest subsequence with limited sum

```
public int[] answerQueries(int[] arr, int[] queries) {  
    Arrays.sort(arr);  
    int n = arr.length, m = queries.length;  
    for(int i=1;i<n;i++){  
        arr[i] += arr[i-1];  
    }  
    int[] ans = new int[m];  
    for(int i=0;i<m;i++){  
        int lo = 0, hi = n-1;  
        while(lo<=hi){  
            int mid = lo + (hi-lo)/2;  
            if(arr[mid]>queries[i]) hi = mid - 1;  
            else{  
                ans[i] = Math.max(ans[i],mid+1);  
                lo = mid + 1;  
            }  
        }  
    }  
    return ans;  
}
```

Sort  $\rightarrow n \log n$

pre  $\rightarrow n$

B.S  $\rightarrow m \cdot \log n$

T.C. =  $O((m+n) \log n)$

[Leetcode 2389]

# Ques:

## Q5: Corporate Flight Bookings

arr = { {1, 4, 10}, {2, 3, 20}, {2, 5, 25} }      n=5

0	1	2	3	4
1	2	3	4	5
10	<del>10</del>	<del>10</del>	<del>10</del>	25
	30	30	35	
	55	55		

1	2	3	4	5
10	<del>20</del>	0	-20	-10
	45			
10	55	55	35	25

[Leetcode 1109]

# Ques:

## Q5: Corporate Flight Bookings

```
public int[] corpFlightBookings(int[][] arr, int n) {  
    int[] ans = new int[n];  
    for(int i=0;i<arr.length;i++){  
        int first = arr[i][0];  
        int last = arr[i][1];  
        int seats = arr[i][2];  
        ans[first-1] += seats;  
        if(last<n) ans[last] -= seats;  
    }  
    for(int i=1;i<n;i++){  
        ans[i] += ans[i-1];  
    }  
    return ans; // 3ms  
}
```

T.C. =  $O(m+n)$

Optimised via Prefix Sum

[Leetcode 1109]

# Ques:

## Q5: Corporate Flight Bookings

*length = m*

```
public int[] corpFlightBookings(int[][] arr, int n) {  
    int[] ans = new int[n];  
    for(int i=0;i<arr.length;i++){  
        int first = arr[i][0];  
        int last = arr[i][1];  
        int seats = arr[i][2];  
        for(int j=first-1;j<=last-1;j++){  
            ans[j] += seats;  
        }  
    }  
    return ans; // 748ms  
}
```

T.C. =  $O(m \cdot n)$

$\rightarrow O(n)$

Brute Force

[Leetcode 1109]

# Ques:

↗ Famous

## Q6 : Subarray sum equals K

arr = { 1, 2, 4, 3, 7, 8, -13 }      K = 7

→ { 1, 2, 4 } , { 4, 3 }, { 7 }, { 8, -13 }      ans = 4

1, 2, 4, 0, 3, -3,

[Leetcode 560]

# Ques:

**Q6 : Subarray sum equals K**

Brute Force Solution

Generating all subarrays  $\rightarrow$  T.C. =  $O(n^2)$   
S.S. =  $O(1)$

**[Leetcode 560]**



**Ques:**

map < ele, freq >

**Q6 : Subarray sum equals K**

Optimised Solution : Prefix Sum & Hashmap

arr = { 1, 2, 4, 3, -3, 0, 7, 8, -1 }      K = 7

pre = { 1, 3, 7, 10, 7, 7, 14, 22, 21 }

(22, 1)  
(14, 1) (21, 1)  
(7, 3)  
(3, 1)  
(1, 1)

int rem = pre[i] - K;

T.C. =  $O(n)$

E.S. =  $O(n)$

count

0

1

2

3

4

7

8

[Leetcode 560]

**Ques:**

$$T.C. = O(n)$$

$$A.S. = O(n)$$

**Q7: Minimum Penalty for a shop**

1 2 3 4  
Y Y N Y

$$0 \text{ hours} \rightarrow 0 + 3 = 3$$

$$1 \text{ hour} \rightarrow 0 + 2 = 2$$

$$2 \text{ hours} \rightarrow 0 + 1 = 1$$

$$3 \text{ hours} \rightarrow 1 + 1 = 2$$

$$4 \text{ hours} \rightarrow 1 + 0 = 1$$

Y Y N Y  
pre(N) 

0	0	0	1	1
---	---	---	---	---

Suf(Y) 

3	2	1	1	0
---	---	---	---	---

  
Y Y N Y

pen 

3	2	1	2	1
---	---	---	---	---

[Leetcode 2483]

# Homework:



**Q : Reducing Dishes**

**[Leetcode 1402]**

◀ **THANK YOU** ▶