Prefix Sum

Lecture-34

Raghav Garg

What is Prefix Sum?

arr = { 1, 4, 5, 3, 2, 7, 6 }

pre = { 1, 5, 10, 13, 15, 22, 28 }

M-I: T.C. = $O(n^2)$ → no. of ops = $1 + 2 + 3 + 4 + \dots n$

M-2: Single pass → T.C. $O(n)$

Ques: Running sum of 1D Array [Leetcode - 1480]

	0	1	2	3
nums =	1	3	6	10

a, b → sum

pre[b] - pre[a-1]

```
for (i = 1 to i = n-1){
    nums[i] = nums[i] + nums[i-1];
}
```

for ex :

	0	1	2	3	4	5	6	7	8
	9	1	8	6	5	2	4	3	10

↳ [9, 10, 18, 24, 29, 31, 35, 38, 48]

Ques: Check if array can be partitioned into 2 continuous arrays of equal sum.

arr = { 1, 2, 3, 4, 5, 6, 7, 8, 9 }

	0	1	2	3	4	5	6	7	8
pre = {	1,	3,	6,	10,	15,	21,	28,	36,	45 }

x

0 to x == x+1 to n-1

pre[x] = pre[n-1] - pre[x+1-1]

$2 * pre[x] = pre[n-1]$

Ques: Product of array except self [Leetcode - 238]

arr = {1, 2, 3, 4}

ans = {24, 12, 8, 6}

product = 24

prefix product

pre = {1, 2, 6, 24}

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

Ques: Product of array except self [Leetcode - 238]

arr = {1, 2, 3, 4}

pre = {1, 1, 2, 6}

suf = {24, 12, 4, 1}

ans = {24, 12, 8, 6} ✓

~~p = 1 * 6~~

p = 1 * 1 * 2 * 4

p *= nums[i]

Concept Involved : For Every Element → product of all other elements

= product of all other in left part

* product of all other in right part

Ques: Product of array except self [Leetcode - 238]

nums = { 4, 2, 5, 3 }

pre = { 1, 4, 8, 40 }

suf = { 30, 15, 3, 1 }

pre[0] = 1

p = nums[0]

p = 4 * 40

suf[n-1] = 1

p = nums[n-1]

p = 3 * 30

Suffix Sum:

arr = { 1, 2, 3, 4 }

pre = { 1, 3, 6, 10 }

suf = { 24, 24, 12, 4 }

Suffix Product:

arr = { 1, 2, 3, 4 }

prep = { 1, 2, 6, 24 }

sufp = { 24, 24, 12, 4 }

ans = { 24, 12, 8, 6 }

Ques: Minimum Penalty for a shop [Leetcode - 2483]

String c = 0 1 2 3 4
Y Y N Y

10th baje shop → yes

11th baje → yes

12 baje → no

13 baje → yes

14 baje → close Penalty = 0 1

0th → band
↓
Penalty - 0, 1, 3 → 3

1st → band kar dete
↓
penalty - 2
0 + 1 + 0 + 1 = 2

2nd hour band kar doon
↓
penalty - 1

3rd hour → penalty - 2

4th hour → 1

Ques: Minimum Penalty for a shop [Leetcode - 2483]

After closing the shop → penalty = no. of Y after that hour

penalty if we close the shop at n^{th} hour

= no. of 'Y' $\geq n$ + no. of 'N' before n^{th} hour

$\geq n$: including & after n^{th} hour

= Y ka suffix sum + N ka prefix

```
      0 1 2 3 4
s =   Y Y N Y
      0 1 2 3 4
pre = 0 0 0 1 1    N
                   Y
suf = 3 2 1 1 0
```

pre = 3 2 1 2 1 minPen = 1

```
// YNYY    // 0
for(int i=0;i<n;i++){
    int count = 0;
    if(customers[i]=='N') count++;
    pre[i+1] = pre[i] + count;
}
```

0 1 2 3 4

Y N Y Y

pre 0 0 1 1 1

suf 3 2 2 1 0

Ques: Reducing Dishes [Leetcode - 1402]

↳ Sorting & Suffix Sum

arr = {4, 3, 2}

1) Chef can prepare the dishes in any order

2) He can also discard some/all dishes

3 x 2

1 x 4 2 x 3

Sort

⇒ 1x4 + 2x3 + 3x2 = 4+6+6 = 16

{2, 3, 4}

↳ = 1x2 + 2x3 + 3x4 = 2 + 6 + 12 = 20

Ques: Reducing Dishes [Leetcode - 1402]

arr = {-1, -8, 0, 5, -9}

Sort ↳ arr = {-9, -8, -1, 0, 5}

⇒ 1x-9 + 2x-8 + 3x-1 + 4x0 + 5x5

⇒ -9 -16 -3 + 0 + 25

⇒ -3 (is bad)

⇒ {0, 5}

⇒ 1x0 + 2x5 = 0 + 10 = 10 (is better)

{-1, 0, 5}

= 1x-1 + 2x0 + 3x5

= -1 + 0 + 15

= 14

Ques: Reducing Dishes [Leetcode - 1402]

$\{-8, -1, 0, 5\}$

$\Rightarrow -8 \times 1 + -1 \times 2 + 0 \times 3 + 5 \times 4 = -8 - 2 + 20 = 10$

- All dishes with +ve satisfaction values must be taken & Some of the dishes with negetive value can be taken.

Ex → $\{-9, 3, 4, 5\}$

$\Rightarrow -9 \times 1 + 3 \times 2 + 4 \times 3 + 5 \times 4$

$\Rightarrow -9 + 6 + 12 + 20$

$\Rightarrow 29 > 26$

$\{3, 4, 5\}$

$3 \times 1 + 4 \times 2 + 5 \times 3$

$= 3 + 8 + 15 = 26$

Ques: Reducing Dishes

[Leetcode - 1402]

arr = {-1, -8, 0, 5, -9}

Sort →

arr = {-9, -8, -1, 0, 5}

Suffix array →

suf = {-13, -4, 4, 5, 5}

idx

Ques: Longest subsequence with limited sum

[Leetcode - 2389] (!Easy)

nums = {4, 5, 2, 1} → n

↓ Sort (O(n log n))

nums = {1, 2, 4, 5}

↓ prefix sum (O(n))

nums = {1, 3, 7, 12}

queries = {3, 10, 21} → m

ans = {2, 3, 4} → m

```
for(int i=0;i<m;i++){
    int len = 0;
    for(int j=0;j<n;j++){
        if(nums[j]>queries[i]) break;
        len++;
    }
    ans[i] = len;
}
```

len = 0 1 2 3 4

[Sorting, prefix sum, Binary Search]

Ques: Longest subsequence with limited sum

[Leetcode - 2389]

→ Sorting & prefix sum

nums = {1, 3, 7, 12}

queries = {3, 10, 21} → m

ans = { } → m

Thank you!