

Sliding Window Lecture-35

Raghav Garg



Utility of sliding window

- Subarray prefix sum? Sliding window? ps, sw dono
- Substrings
- Largest / Smallest sum
- In a window/subarray of given size 'k'



Ques: Maximum sum Subarray of size k

arr =
$$\{2, 1, 2, 5, 8, 4, 9, 3, 63\}$$
 $n = 9$
 $\{4, 1, 2\}$, $\{1, 2, 5\}$, $\{2, 5, 8\}$, $\{5, 8, 4\}$, $\{8, 4, 9\}$
 $\{4, 9, 3\}$, $\{4, 3, 6\}$



Ques: Maximum sum Subarray of size k

```
Let us use the sliding window algorithm
          0 1 2 3 4 5 6 7 8
arr = \{4, 1, 2, 5, 8, 4, 9, 3, 63\} n = 9
   blue sum = gneen sum +8-7
  curr window = prev window + arr[i] - arr[i-1]
       Cum
```

$$arr = \{2, 1, 2, 5, 8, 4, 9, 3, 63\}$$

```
while ( j 4 n) {
int curr Sum = prev Sum + arr [i] - arr [i-1];
   if (maxCum < curr Sum) {
     maxcum = curscum;
      max Idx = i;
   prevsum = curssum;
```



Ques: Grumpy Bookstore owner

[Leetcode - 1052]

```
// [1,0,1,2,1,1,7,5] - Customers
// [0,1,0,1,0,1,0,1] 3 min grum?
// [0,1,0,1,0,0,0,0]
                              modified
    00000
Brute force: try every window
           \neg \tau \tau.c. = O(n-k)^{\tau} \eta
```

```
Sliding window algo
# Hint: Find that window
         that has most logs
         of Satisfaction
        3 1= the window
           with the least
Satisfaction
```



```
Sliding Window Algo
```

```
// [1,0,1,2,1,1,7,5] arr
// [0,1,0,1,0,1,0,1] 3 min
```

maxloss maxIax

luss of satisfaction, -> Lus

only if
$$gau[i-1]==1$$

```
SKILLS
```

```
customers =
[9,10,4,5]
grumpy =
[1,0,1,1]
minutes =
```

```
int prevLoss = 0;
for(int i=0;i<k;i++){
    if(grumpy[i]==1) prevLoss += arr[i];
}
int maxLoss = prevLoss;
int maxIdx = 0;
int i = 1;
int j = k;</pre>
```

```
customers =
[9,10,4,5]
grumpy =
[1,0,1,1]
```

```
while(j<n){
    int currLoss = prevLoss;
   if(grumpy[j]==1) currLoss += arr[j];
    if(grumpy[i-1]==1) currLoss -= arr[i-1];
    if (maxLoss<currLoss) {
        maxLoss = currLoss;
        maxIdx = i;
    prevLoss = currLoss;
    <u>i</u>++;
// filling 0s in the grumpy array window
for(int i=maxIdx;i=n;i++){
  grumpy[i] = 0; \rightarrow \downarrow \angle max[d \times k]
// sum of satisfaction
int sum = 0;
for(int i=0;i<n;i++){
    if(grumpy[i]==0) sum += arr[i];
return sum;
```

```
Currloss = 9098 | Skills

previous = 9090 | Maxious = 9

maxious = 9

i = 1289
```



Ques: First negative number in every window of size k

orr =
$$\{2, -3, 4, 4, -7, -1, 4, -2, 6\}$$
 K=4

P

i = start o

while $(j < n) \le // n$ iteration is

the $(p > i)$ and $(i) = arr[p]$;

else $\{for[p = i; p < = j; p + t) \le // k$ iteration if $(arr[p] < 0)$ break;

3

i = start o

i = arr[p];

i = start of window

while
$$(j < n) \le // n$$
 iterations

if $(p > i)$ ans $[i] = arr[p]$;

else $\{for[p=i;p<=j;p+t) \le // k$ iterations

if $(arr[p] < 0)$ break;

ans $[i] = arr[p]$;

Ques: Minimum size subarray sum [Leetcode - 209]

4) I have to find the window size

$$arr = \{2, 3, 1, 2, 4, 3\}$$
 tanget = 7

$$n + n-1 + n-2 \dots 1 = n(n+1) \Rightarrow T \cdot C = O(n^2)$$

nums =
$$\{1, 2, 4, 6, 3, 4, 3\}$$
 target = 10

Sum = 0/34 1312 10 89137 | Sum += arr[j]

len = 0 4 82 3 3 | 107 | while (sum >= ta

ninlen = INI=MAX 4 3 2

Sum -= arr

10 T.C. = O(n) ムもno -> 42 n

while (j-m) 4 107 While (Sum>= target) (len = j-i+1; minlen = min(minlen, len); Sum -= ary[i]; 3

Ques: Max consecutive ones III

mems =
$$\{2,0,1,1,0,0,0,1,1,1,0,0\}$$

[Leetcode - 1004]

3) clse

Den = j-i

maxlen =
$$max(maxlen, len)$$

Ques: Max consecutive ones III

1) nums
$$GiJ = 1$$
 $\rightarrow j+t$

2) numb
$$G = D$$

[Leetcode - 1004]



```
int zeroPos = -1;
int i = 0;
int j = 0;
int maxLen = 0;
int count = 0;
while(j<n){
    int prev = zeroPos;
    if(nums[j]==0){
        count++;
        zeroPos = j;
    if(count<=1) j++;
    else
        maxLen = Math.max(maxLen,j-i);
       i = prev + 1;
        count--;
maxLen = Math.max(maxLen,j-i);
return maxLen-1;
```

'ay of l's after deleting one [Leetcode - 1493]

```
maxlen = 0 \times 6

count = 0 \times 2 \times 2 

prev = 10

zeroPas = -10
```

Ques: Subarray Product Less than K [Leetcode - 713]

no-of subarrays nikalne hai

$$nums = \{10, 5, 2, 6\}$$

pro
$$= num s LiJ$$
while $(pro <= K) <$
count $+= (j-i)$

$$K=100$$

$$count = 0.25$$

$$product = 1.170 \text{ Solve } 1660$$

```
if(k<=1) return 0;
int n = nums.size();
int i = 0;
int j = 0;
int count = 0;
int product = 1;
while(j<n){
   product *= nums[j];
   while(product>=k){
       count += (j-i);
       product /= nums[i];
        <u>i</u>++;
```

$$\{10, 5, 19, 4\}$$
 $K=100$

while lizn)?



Thank you!