↳ you are given 2 arrays, A & B

A, B } ⟶ sorted in non decreasing order

Count the number of pairs (i, j) for which $a_i = b_j$

A (i) B a[i] = b[j]

$1 \le n, m \le 10^5$

$1 \le a_i, b_j \le 10^9$

A :- 1 1 3 3 3 5 8 8

B :- 1 3 3 4 5 5 5 5

i i i i i i

j j j j

2 3 3 0 1 1 1

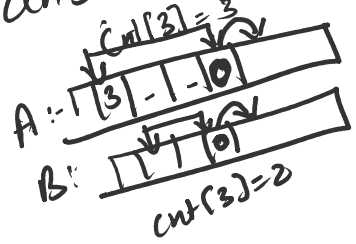
Ans = 11

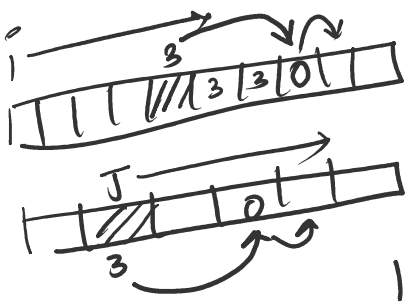
---

map cnt1[x] 1 $10^9$

cnt2[x]

ans = cnt1[e] × cnt2[e]




cnt1[3] × cnt2[3]

cnt1[3] = 3

cnt2[3] = 2

1 $10^9$

Kadanes Algo

You are given an array of size n & you have to find the maximum sum subarray

$1 \le n \le 10^6$

$-10^9 \le A[i] \le 10^9$

[3 4]

1 -5 3 1

4

n → max

n

```
sum=0, ans=0
for(int i=0; i<n; i++)
{
    sum += a[i]
    if(sum<0)
        sum=0
    ans=max(ans, sum)
}
```

max sum of any subarray

```
sum=0, ans=0
l=0 r=0, ansl, ansr
for(int i=0; i<n; i++)
{
    sum += a[i]
    r=i
    if(sum<0)
        l=i+1, sum=0;
    if(ans < sum)
    {
        ans = sum
        ansl = l;
        ansr = r;
    }
}
```

| sum=3 | 4 |
|---|---|
| r=2 | 3 |
| ans=3 | 4 |
| l=2 | 2 |
| r=2 | 3 |

1 -5 3 4

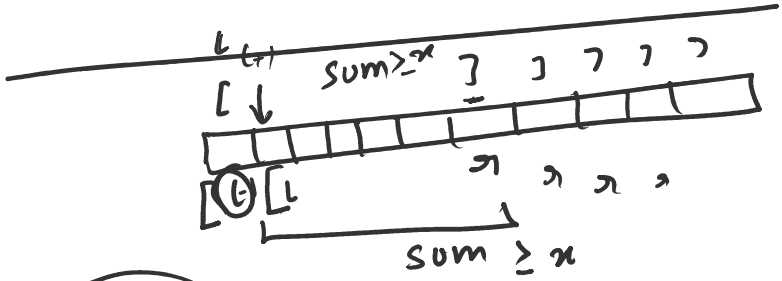
sum=1
r=0
ans=1
ansl=0, ansr=0

Given an array of n numbers
You have to find how many
subarrays exist with sum ≥ x

$1 \le n < 10^6$

$1 \le x \le 10^{18}$

$1 \le A[i] < 10^9$




Sum ≥ x

ans = (n - x) ← 0 indexed

```
Sum -= a[l-1]
while(sum < x) <-- x<=n
{
   x++
   sum += a[x]
}
        if (sum ≥ x)
           ans += (n-x);
```

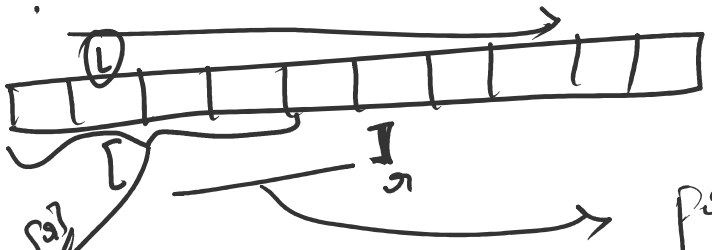
Count Subarrays with Sum

Q Count subarrays with
Sum ≥ x if $-10^9 \le A[i] \le 10^9$
$n \le 10^5$

set

set

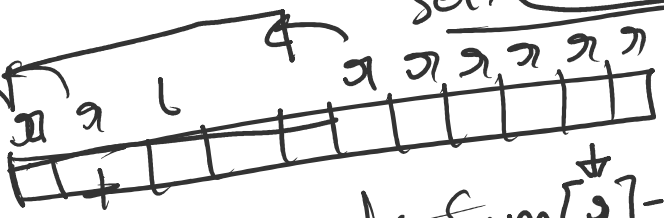
$n \le 10^5$



prefixSum[r] - x

prefixSum[r] - prefixSum[l-1]

set / multiset → storing in sorted order



$\log n$

$Sum(l:r) = prefixSum[r] - prefixSum[l-1] \ge x$

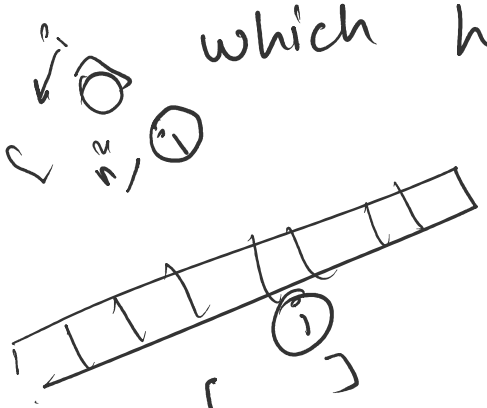
$sum \ge x$

$prefixSum[l-1] \le prefixSum[r] - x$

① nlogn

---

you are given an array, you have to find the subarray which have the maxelem - minelem $\le K$



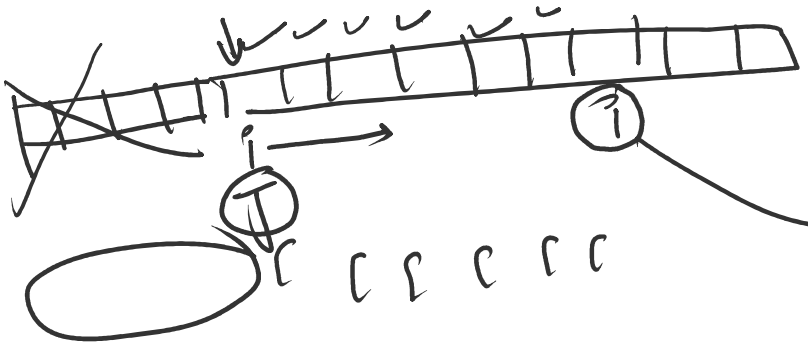
$n < 10^5$

$1 \le K \le 10^{18}$

$1 \le A[i] \le 10^{18}$

for (int i=0 ; i < N; i++)

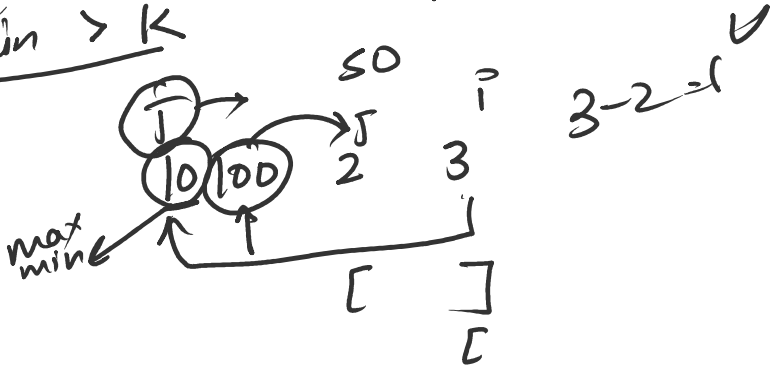
(i - j + 1)



ans += (i - j + 1)

multiset

max − min > k



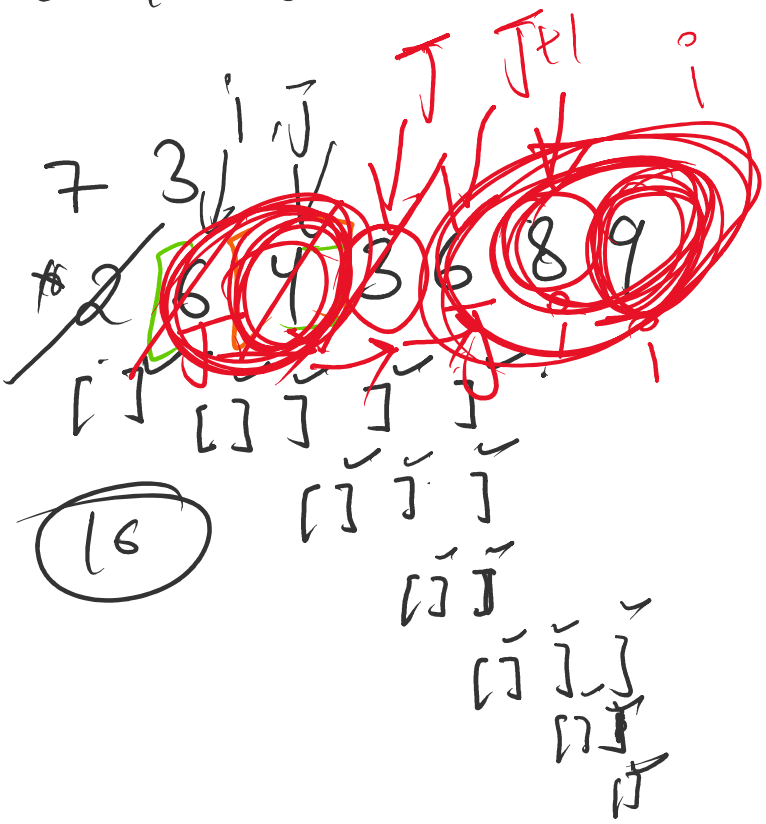

i - j + 1

```
multiset<int> mst
ans = 0
for (int i=0, j=0; i<n; i++)
{
    mst.insert(a[i])
    while (mst_max - mst_min > k)
    {
        remove a[j] → mst
        j++
    }
    ans += i - j + 1
}
```

100 90 80 70 60

100 90 80 70 60

[6]




ans = 0

min = 2

max = 6

(2)

ans = 1

ans = 2

= 4

+3

6 - 4

8 3

7

+4

11

+2

13

+3

16