⊛ Carry Forward

1) Count the pairs "ag"

Given a string, calculate the no. of pairs (i, j) : $i < j \;\&\&\; s[i] == 'a' \;\&\&\; s[j] == 'g'$

0	1	2	3	4	5	6	7
b	a	a	g	d	c	a	g

$(1,3)$, $(1,7)$, $(6,7)$, $(2,3)$, $(2,7)$ → 5

I) BF

```
cnt = 0;
for (i = 0; i < N; i++) {
    for (j = i+1; j < N; j++) {
        if (s[i] == 'a' && s[j] == 'g') {
            cnt++;
        }
    }
}
return cnt;
```

TC: $O(N^2)$

SC: $O(1)$

```
cnt = 0;
for (i = 0; i < N; i++) {
    if (s[i] == 'a') {
        for (j = i+1; j < N; j++) {
            if (s[j] == 'g') {
                cnt++;
            }
        }
    }
}
return cnt;
```

TC: $O(N^2)$

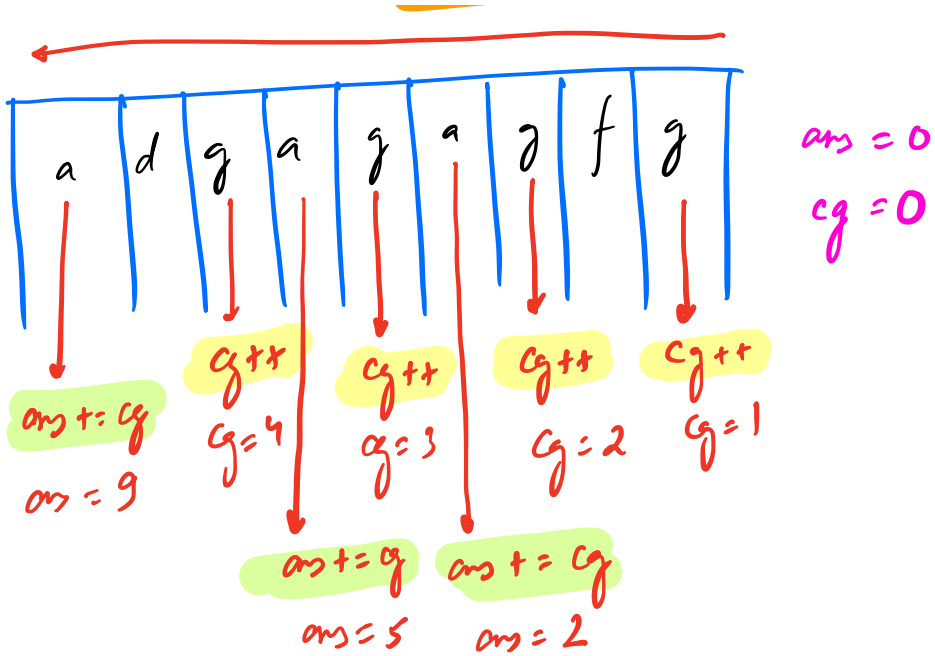
SL = $O(1)$

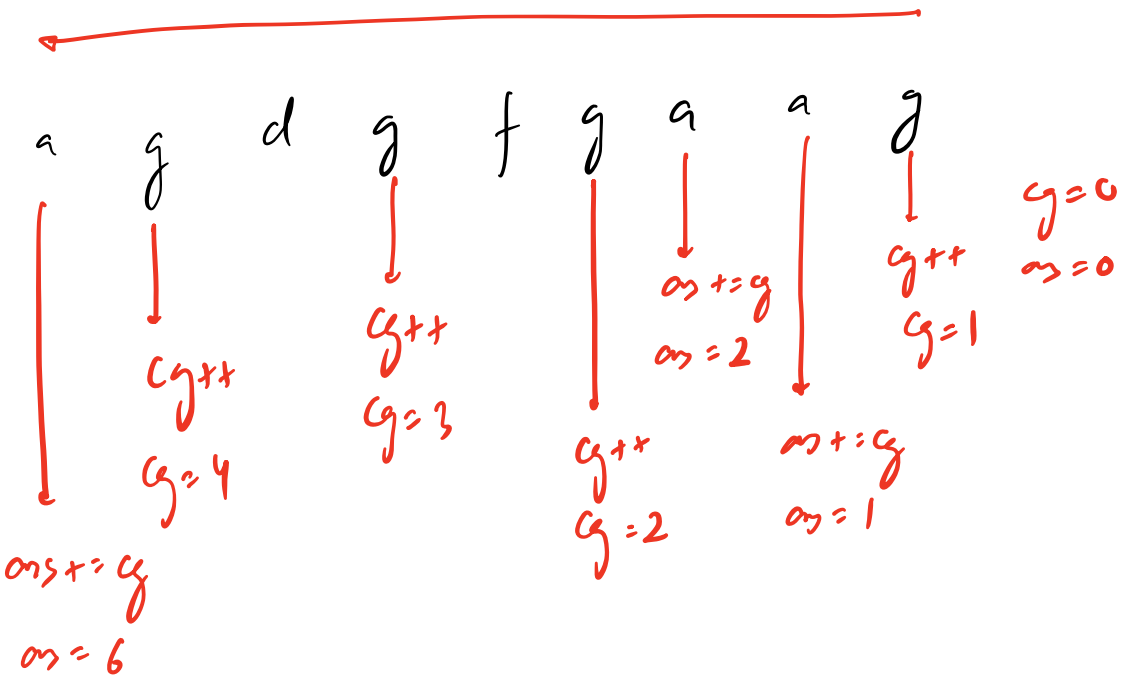
Worst case: "a a a a ——— a a"

	0	1	2	3	4	5	6	7	8
S =	a	d	g	a	g	a	g	f	g

4 + 3 + 2 ⟶ 9 ANS

Obs: for every 'a', count the no of 'g' on the right side of it!


a d g a g a g f g
ans = 0
cg = 0
ans += cg
ans = 9
cg++
cg = 4
cg++
cg = 3
cg++
cg = 2
cg++
cg = 1
ans += cg
ans = 5
ans += cg
ans = 2

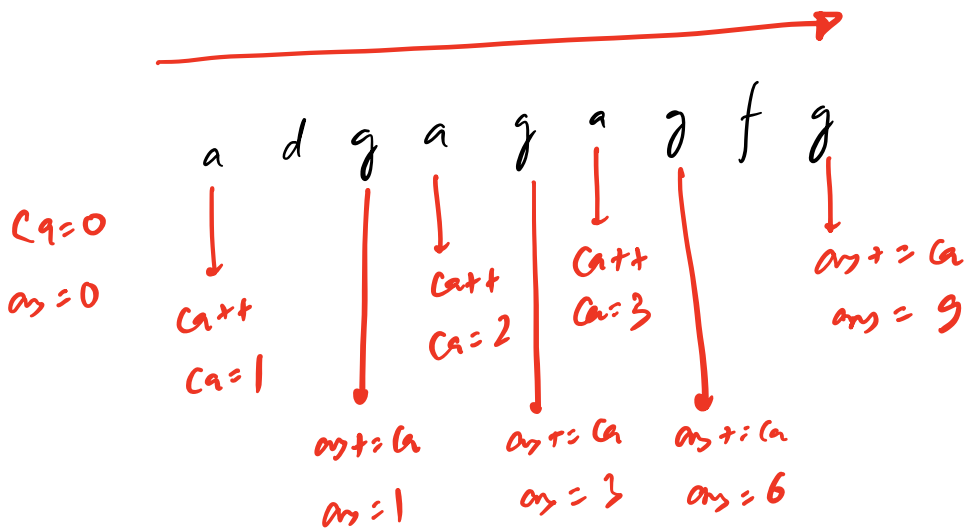

a g d g f g a a g
cg = 0
ans = 0
ans += cg
ans = 6
cg++
cg = 4
cg++
cg = 3
cg++
cg = 2
ans += cg
ans = 2
ans += cg
ans = 1
cg++
cg = 1

```
cg = 0, ans = 0
for ( i = N-1; i >= 0; i --) {
    if ( s[i] == 'g') {
        cg++;
    }
    else if ( s[i] == 'a') {
        ans += cg;
    }
}
ret ans;
```

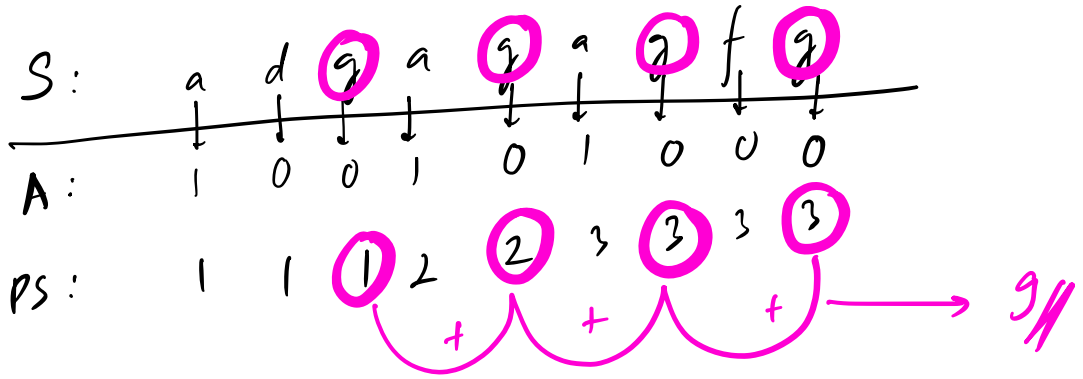
N

TC : O(N)

SC : O(1)

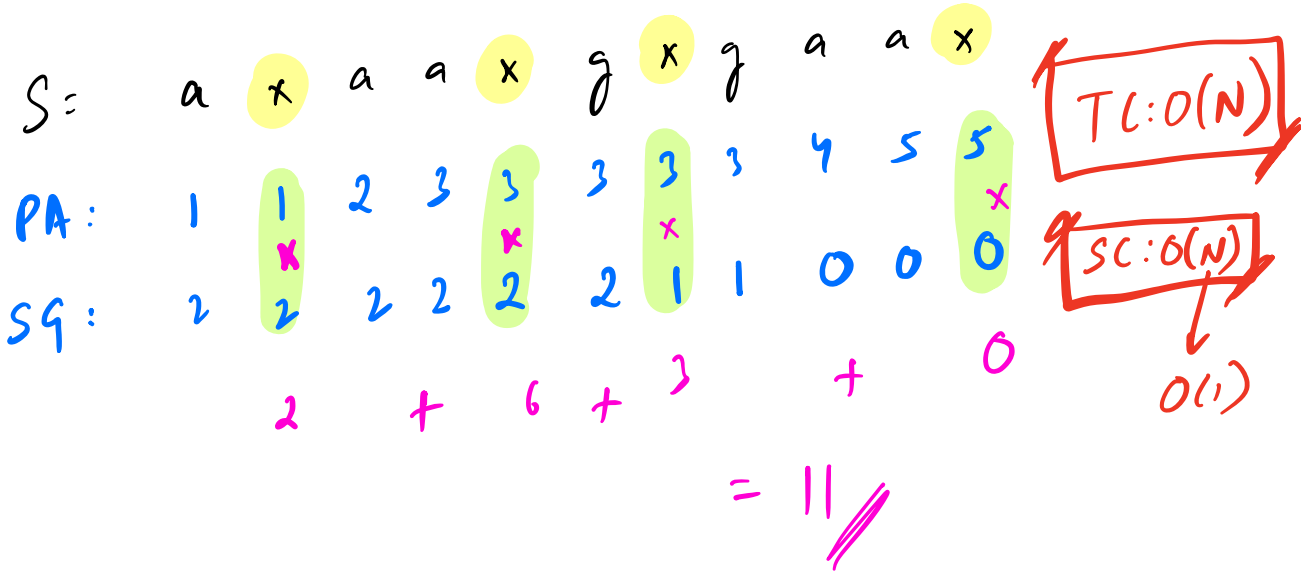


III Prefix Sum



Q "a x g" (i, j, k) $i < j < k$

$S[i] = 'a'$, $S[j] = 'x'$, $S[k] = 'g'$

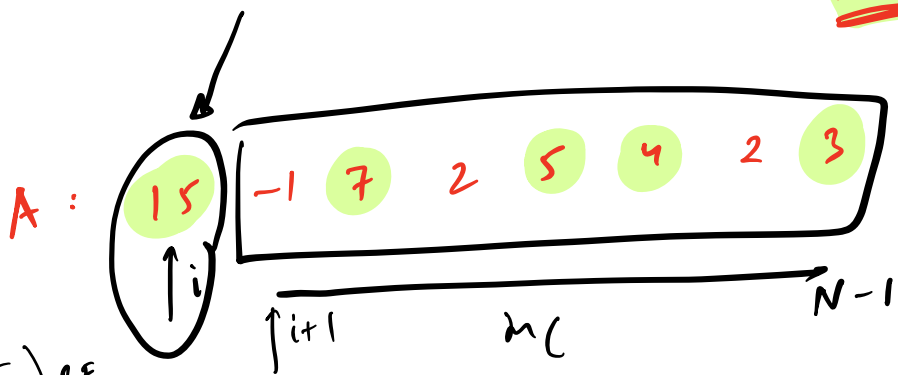


Q Given an array A.

Find the no. of leaders in the array!

An element is a leader if it is greater than all the elements on it's right side!

NOTE: A[N-1] is ALWAYS a leader!



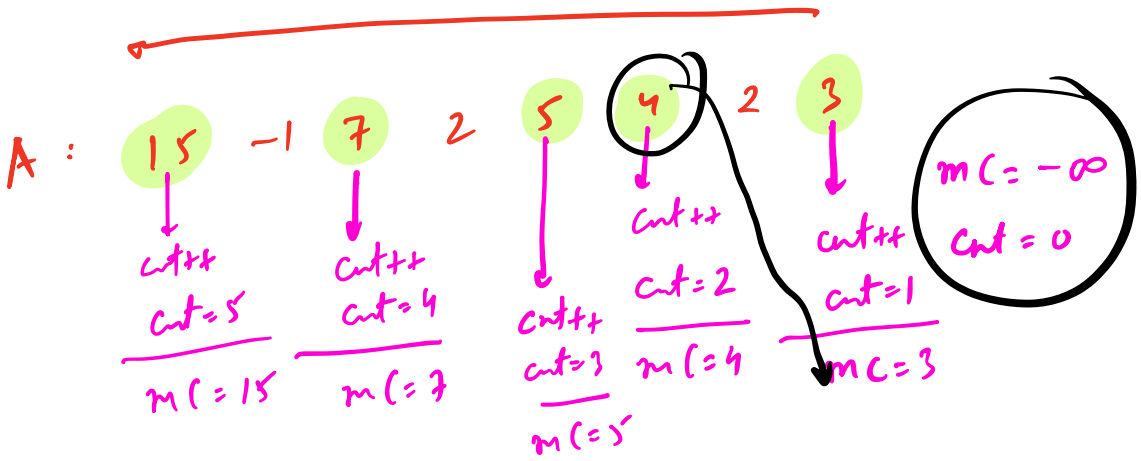
I) BF

```
cnt = 0;
for( i = 0; i < N; i++) {
    mC = -∞;
    for( j = i+1; j < N; j++){
        if( A[j] > mC ){
            mC = A[j];
        }
    }
    if(A[i] > mC){
        cnt++;
    }
}
ret cnt;
```

TC = O(N²)

SC = O(1)

II




```
mC = -∞, cnt = 0
for(i = N-1; i >= 0; i--){
    if(A[i] > mC){
        cnt++;
        mC = A[i];
    }
}
ret cnt;
```

TC = O(N)

SC = O(1)

Q Closest Min Max →

Given an array, find the length of the smallest subarray which contains both the MIN & MAX of the array!

	0	1	2	3	4	5	6	7	8	9
A:	1	2	3	1	3	4	6	4	6	3

→ 4 //

I) BF

1. Find MIN → O(N)
2. — MAX → O(N)
3. Consider every S.A. [L, R) → N^2

   → check if it contains MIN → O(N)

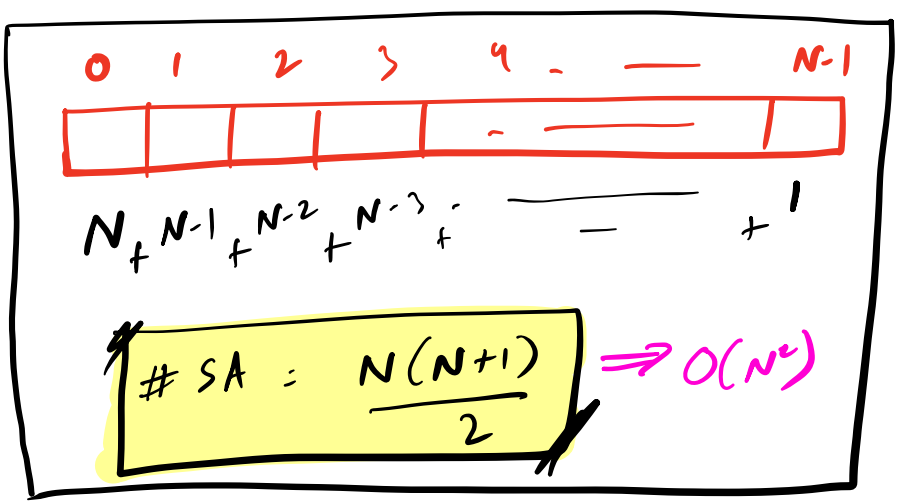
   → — MAX

   → YES: ANS = MIN(ANS, R-L+1)

   → NO: X

$O(N^3)$

TC: $O(N^3)$



A: 8 8 8 8 ⟶ ANS = 1

if(MIN == MAX)
ANS = 1

A: 2 2 6 4 5 1 5 2 6 4 1

MAX MIN MAX MIN

Obs

1. _MIN _ MAX _MIN_ X

_MAX _ MAX _MIN X

the ANS S.A. would contain
EXACTLY 1 occ of both MIN & MAX !

2. _MIN _ MAX _ X

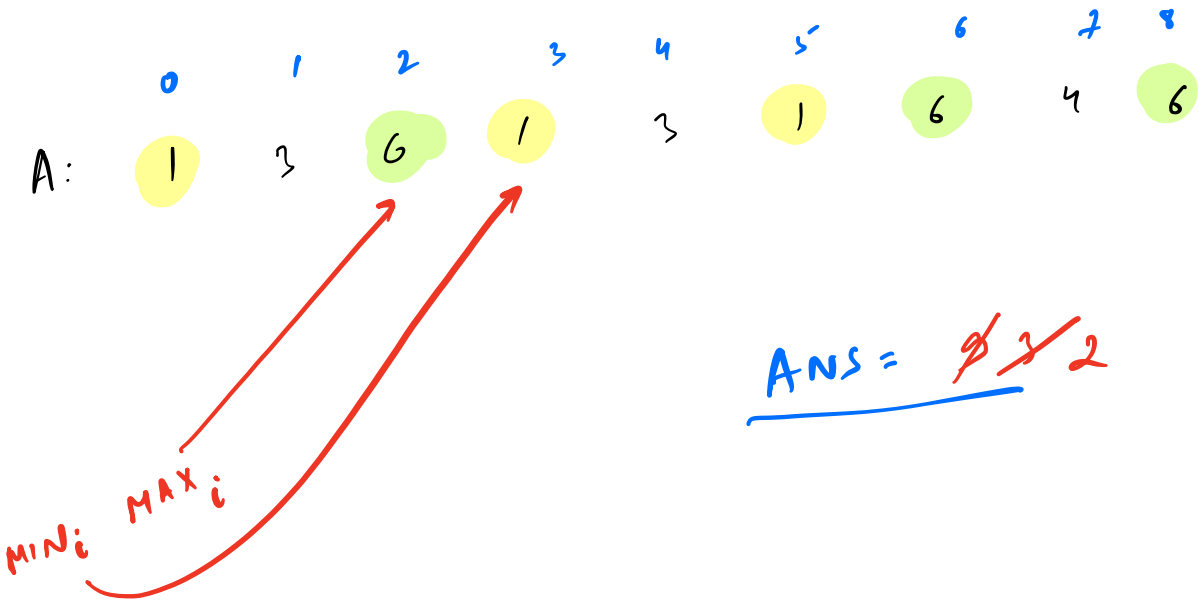
↓

MIN _ MAX

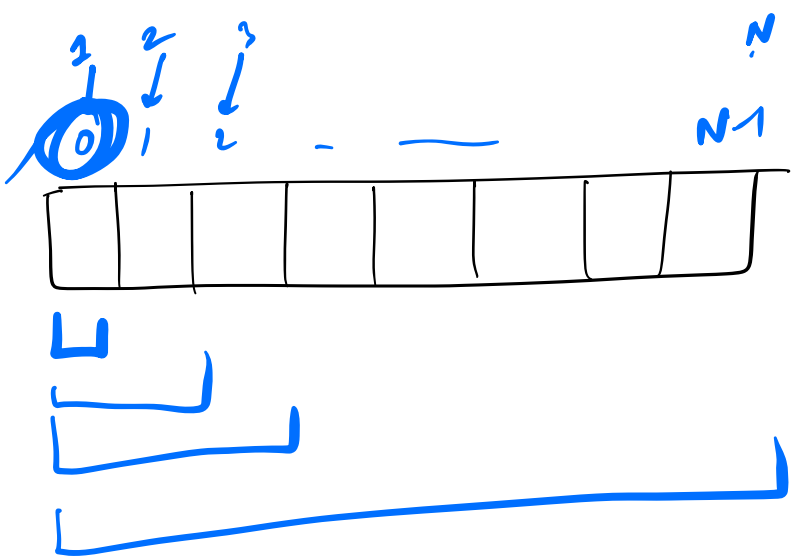
the ANS S.A. would contain
MIN & MAX at boundaries !

↳ Gr I : MIN MAX

↳ Gr II : MAX MIN



ANS = ~~0~~ ~~1~~ 2

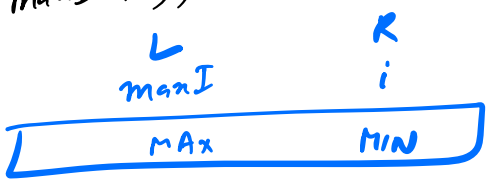


1. Find MIN → N
2. Find MAX → N
   if (MIN == MAX) ret 1;
3. minI = -1, maxI = -1, ans = N

```
for (i = 0; i < N; i++) {
    if (A[i] == MIN) {
        if (maxI != -1) {
            ans = min(ans, i - maxI + 1);
        }
        minI = i;
    }
    else if (A[i] == MAX) {
        if (minI != -1) {
            ans = min(ans, i - minI + 1);
        }
        maxI = i;
    }
}
ret ans;
```

i, j

N x O(1)



N + N + N

= 3N

TC : O(N)

SC = O(1)