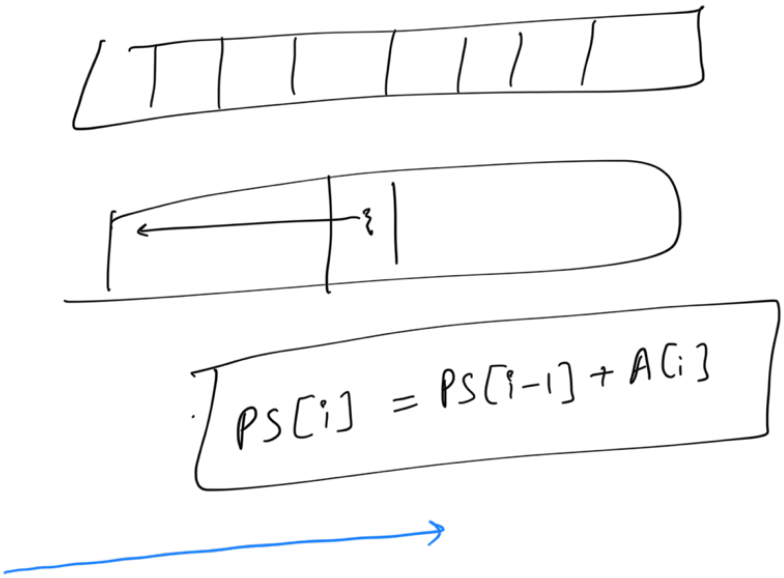
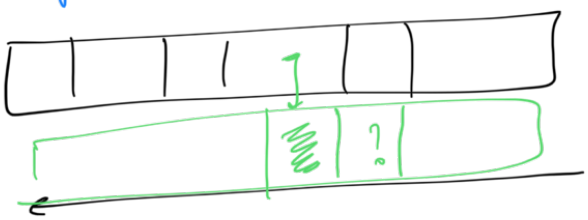
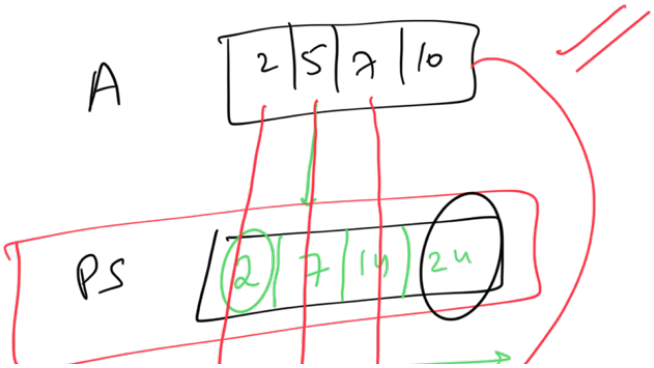
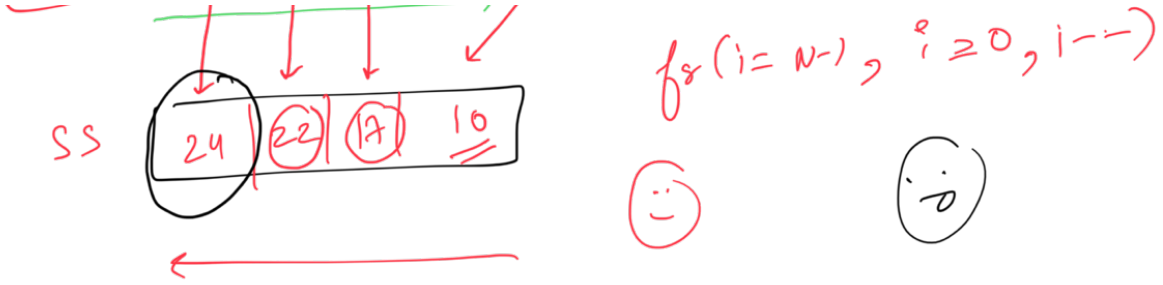
Arrays - Carry Forward

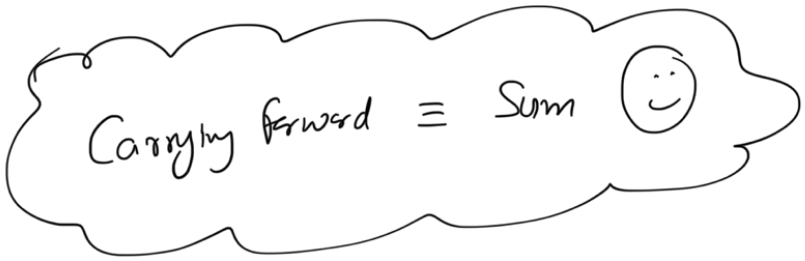


Suffix Sum



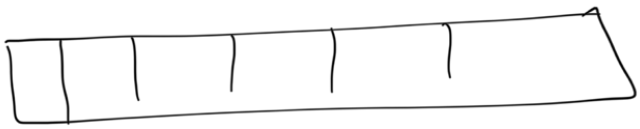






0

A =



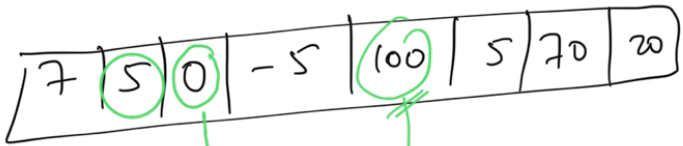
Create another array, A'

where A'[i]

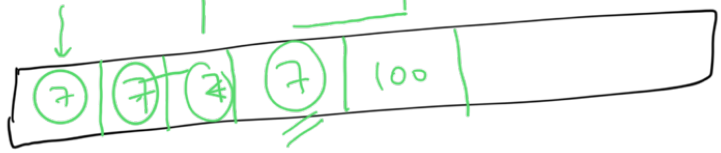
↓

maximum value of array from index 0 to i

A =

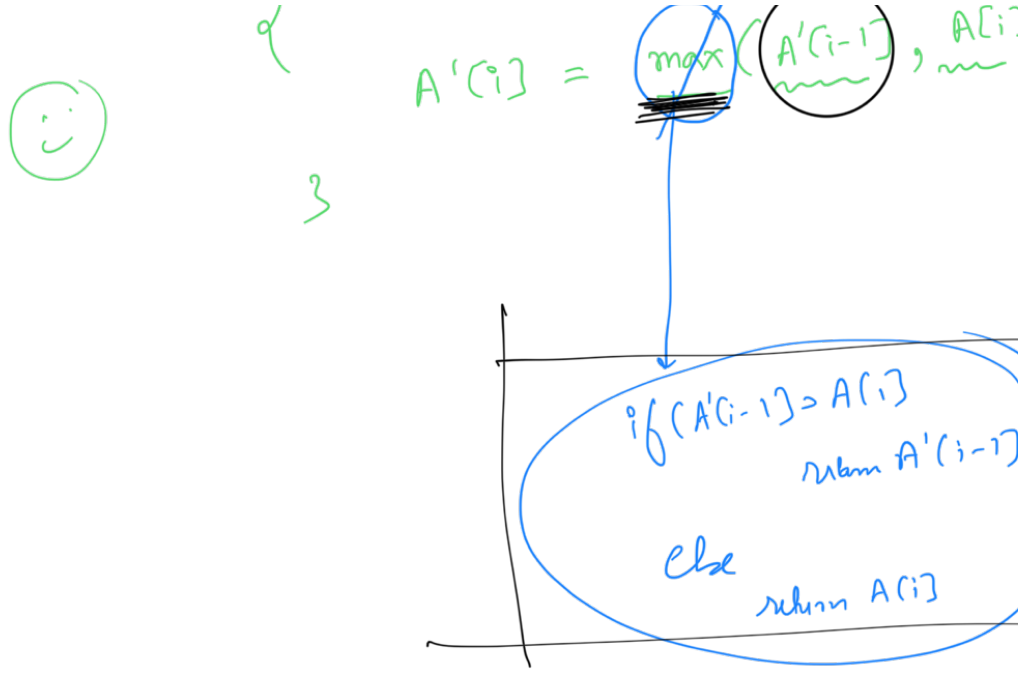


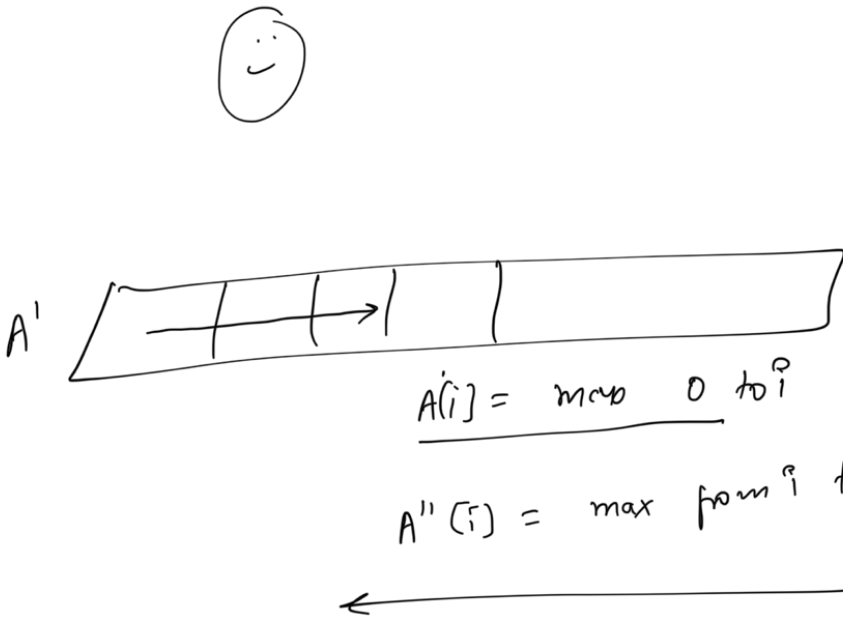
A' =

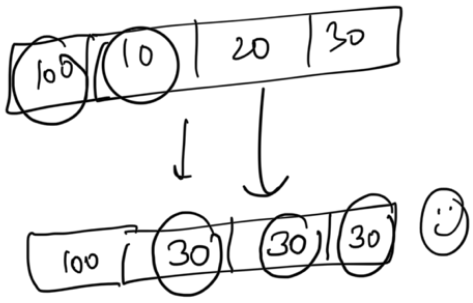


A'[0] = A[0]

for (i = 1, i < N, i++)


{
A'[i] = max(A'[i-1] , A[i]
}
if (A'[i-1] > A[i]
return A'[i-1]
else
return A[i]

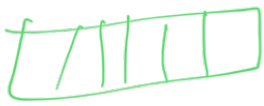

A'
A'[i] = max 0 to i
A''[i] = max from i to N-1


100
10
20
30
100
30
30
30

$LeftMax[i] = Max(LeftMax[i-1], A[i])$

$A' \equiv$ Max from 0 to i

$$A'[i] = \begin{cases} A[i] & i = 0 \\ \max(A'[i-1], A[i]) & i \neq 0 \end{cases}$$



$A' \equiv$ Suffix sum Array

$$A'[i] = \begin{cases} A[i] & i = N- \\ A'[i+1] + A[i] & i \neq \end{cases}$$

---

(0) Medium

You are given a string of lower case characters

g g a g a g a a a g g a g a g g g

Return the number of pairs (i,j) where i<j

where

s[i] = 'a'

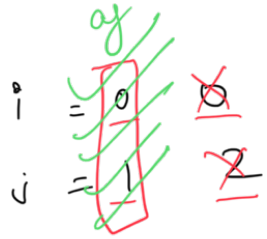
s[j] = 'g'

Ex

i, j

i → 0 to 2

j → 0 to 2



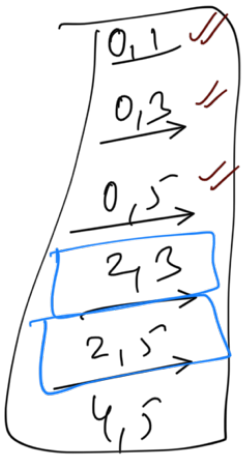
i = 1

j = 2

1

i = 2

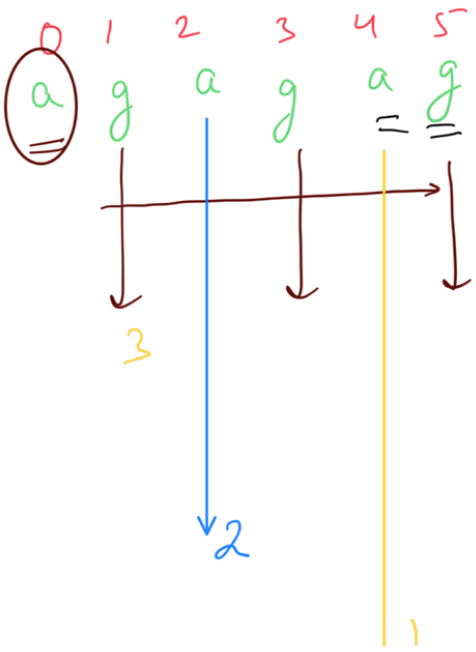
j =



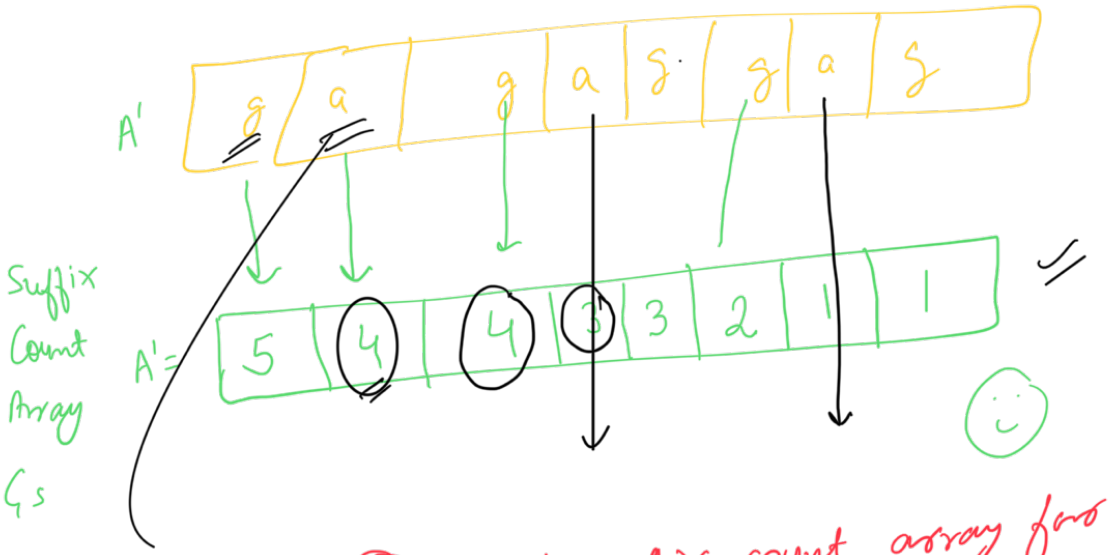
Count=0
for (i=0; i<N, i++) {
for (j=i+1, j<N, j+
{
if (a[i]=='a'
&&
a[j]=='g')
Count ++
}
}

$O(N^2)$

Count



3+2+1 = 6



4
3
1

8

(1) Create suffix count array for g's

T.C = O(N)

S.C = O(N) auxi

(2) Traverse array from 0 to N-1

stop at evy a

and find count

form SC A

O(N) Tim

O(1) spac

O(N) Tim

O(N) extra spac

A → A' → COUNT

Answ

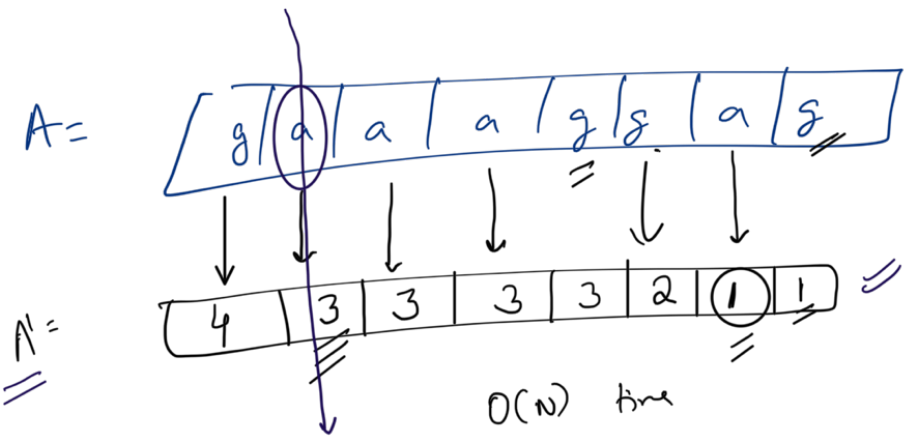
suffix count of g's

A'[i] =

To create auxiliary array

if (A[N-1] == 'a') | i = N-1
0
else if A[N-1] == 'g'

if (A[i] == 'g') | i ≠ N-1
A'[i] = A'[i+1] + 1
else if (A[i] == 'a')
A'[i] = A'[i+1]



answer = 0

start ← for (i=0, i < N, i++)

{

if (s[i] == 'a')

{

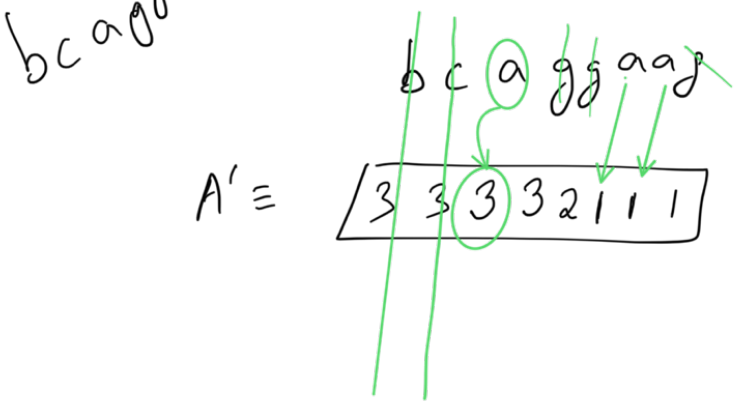
ans = ans + A'[i]

}

}

(8)

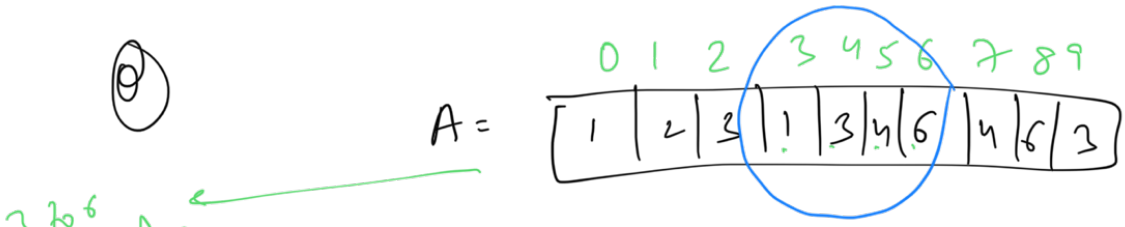
bcaggaag



3
1
1
5

---

(9)



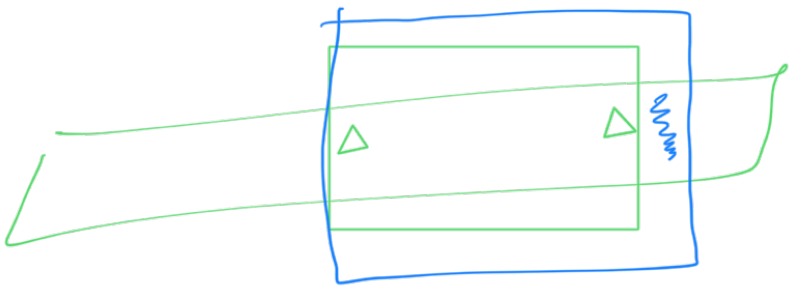
3 to 6

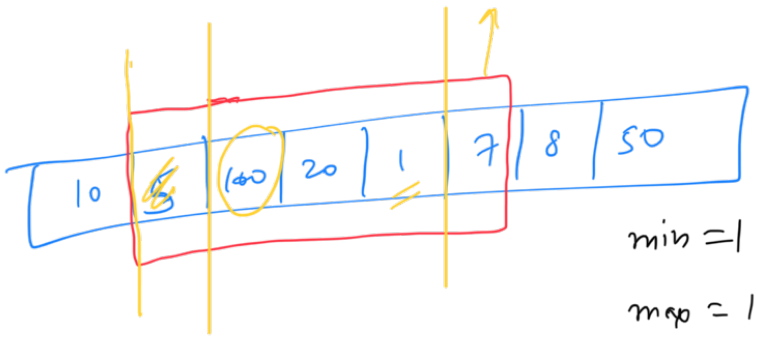
4

Return the smallest length of the subarray which contain both the minimum and the maximum number

smallest = 1

length = 6



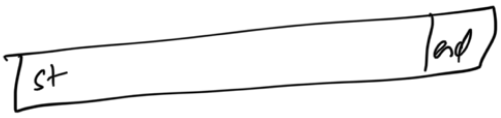


Obs: RoI must end with min and max

Region of Interest

Proof: Let's assume my RoI doesnot end with min and max

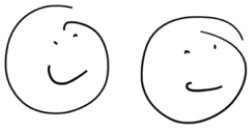
Then



↓

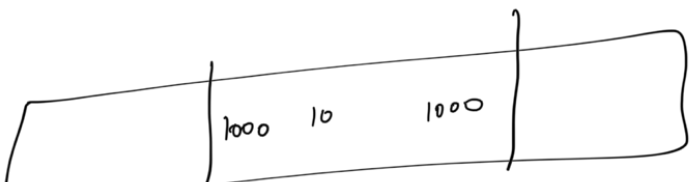
you can keep on shuddly the extremes without loosing the min and max

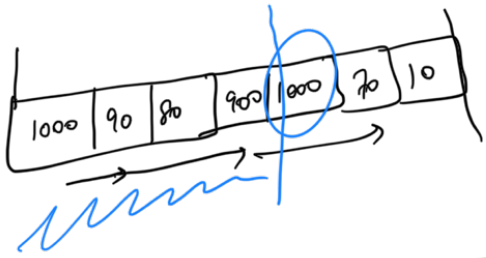
By way of contradiction ← BWOC



10 min

1000 m





A =

1000	50	60	10	20	1000	20	10
0	1	2	3	4	5	6	7

A =

2	2	6	4	5	0	5	2	6	4	1

3

4 4

(3) ☆

min = 1

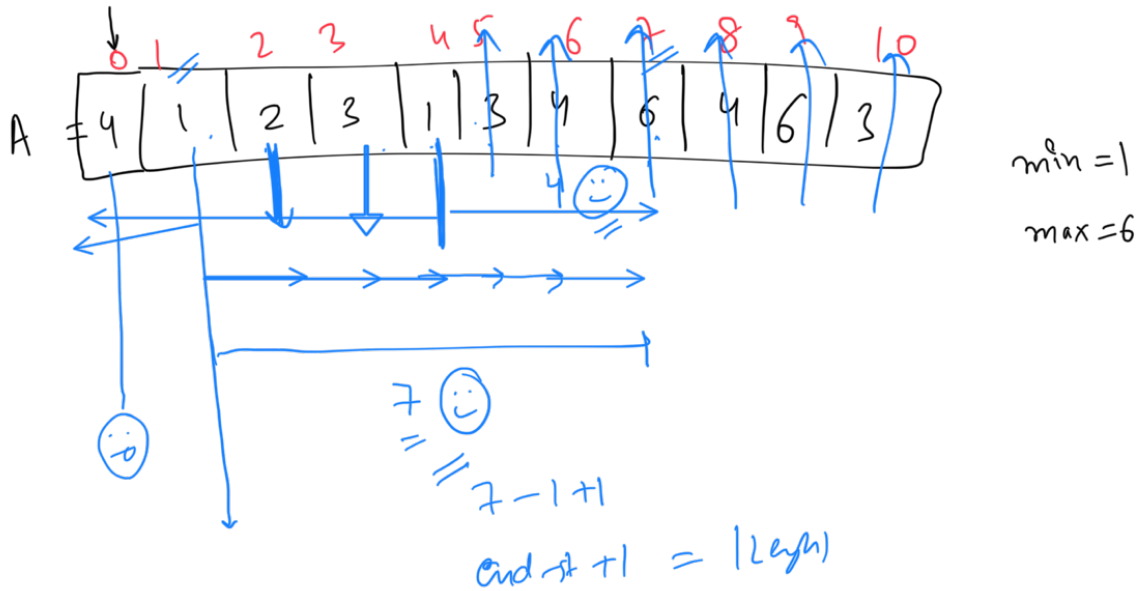
max = 6

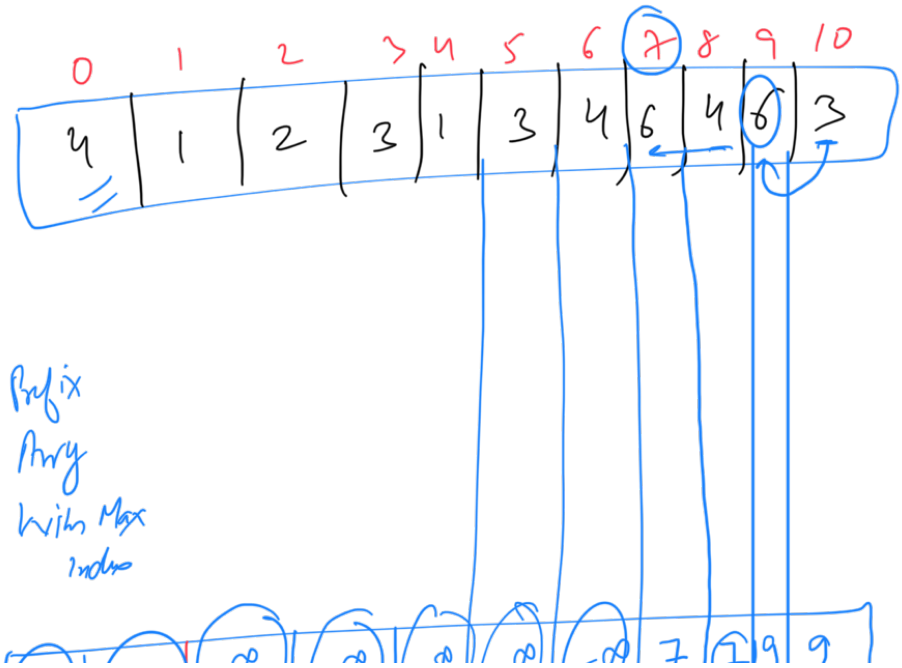
O(N²)

for (st)

for (end)

end - st + 1


A = 4 1 2 3 1 3 4 6 4 6 3
min = 1
max = 6
7 - 1 + 1
end - st + 1 = length

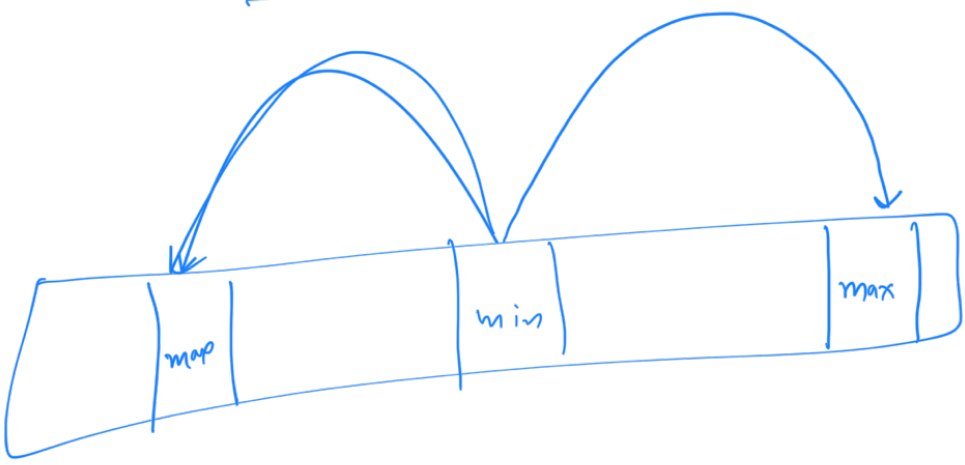

4 1 2 3 1 3 4 6 4 6 3
Prefix Array with Max index

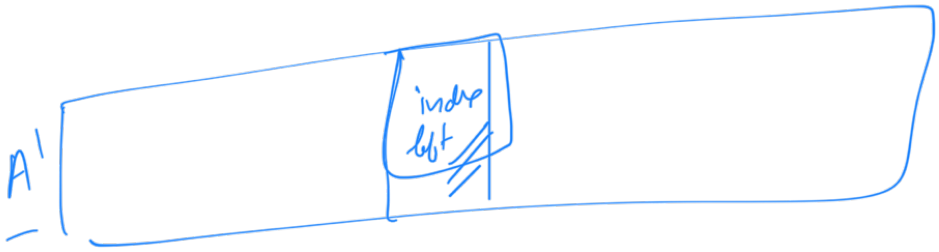
O(n) time

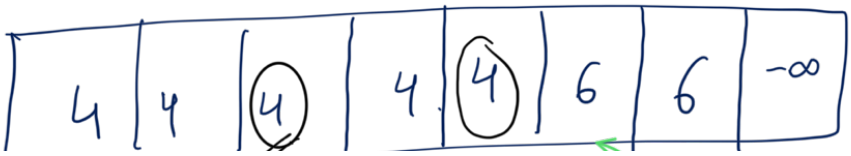
Find max

Find min

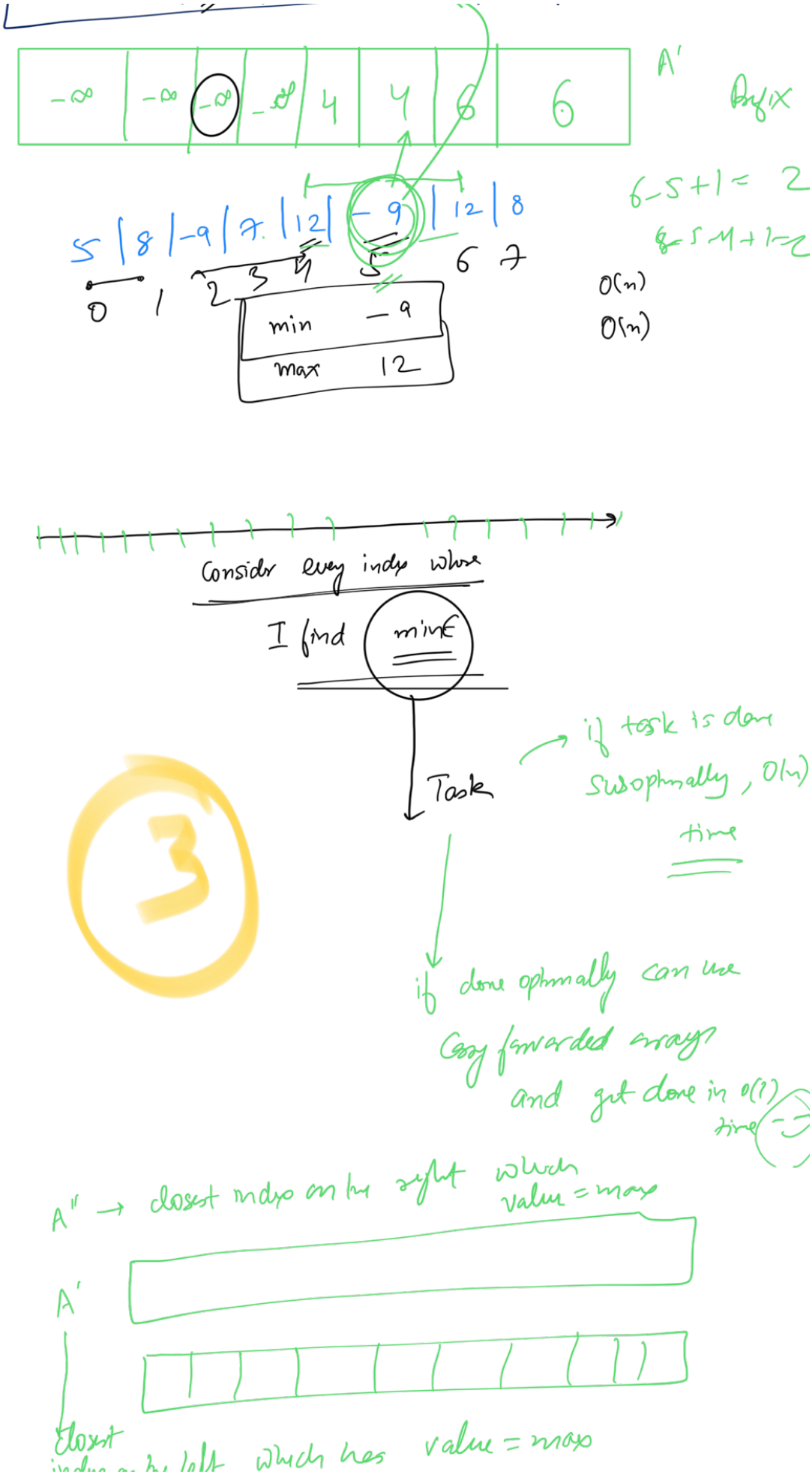
A' = closest index on the left which has the max value







A" = Suffi


A'
Prefix
6-5+1 = 2
8-5+1=2
O(n)
O(n)
min -9
max 12
Consider every index where
I find minE
Task
if task is done suboptimally, O(n) time
if done optimally can use
Easy forwarded arrays
and get done in O(1) time
A'' → closest index on the right which value = max
A'
Closest index on the left which has value = max

(A) ← int getMax(A)

{

max = −∞

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

if (A(i) > max)

max = A(i)

return max

}

(B) ← getMin()

(C) ← Create A'

if A(0) == maxE

A'(0) = 0

else

A'(0) = −∞

for(i=1; i<N; i++)

{

if (A(i) == maxE)

A'(i) == i

else $A'(i) = A'(i-1)$

}

(D) ← creat A''

if $A(N-1) == \max \in$

$A''(N-1) = N-1$

else $A''(N-1) = -\infty$

for ($i = N-2$, $i \geq 0$; $i--$)

if $A(i) == \max \in$

$A''(i) = i$

else $A''(i) = A''(i+1)$

}

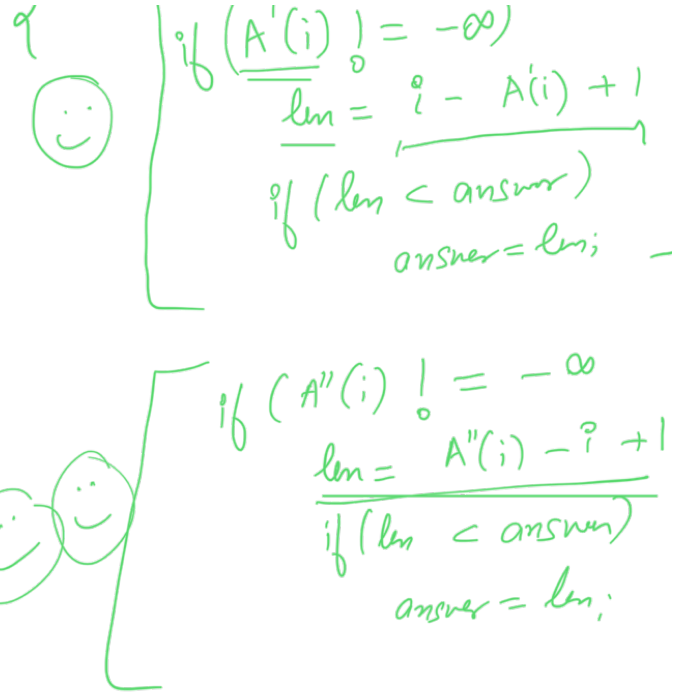
(E) ⟶ Traverse and explore any min ∈

ans = $+\infty$

for ($i=0$, $i < N$, $i++$)

{ if ($A(i) == \min \in$)

{



A = [ ]

Situation where again and again

you have to traverse

to find out any boiled information

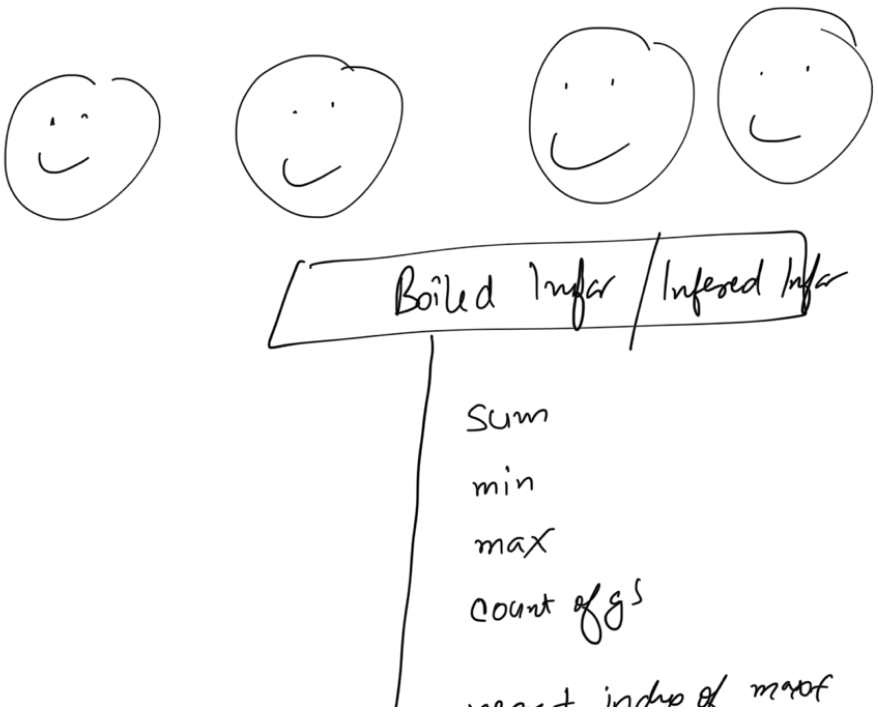
then the idea of conj's forward

makes sense

Then STOP traversing

and think about stay most

boiled info


Boiled Infar / Infused Infa
sum
min
max
count of gs