

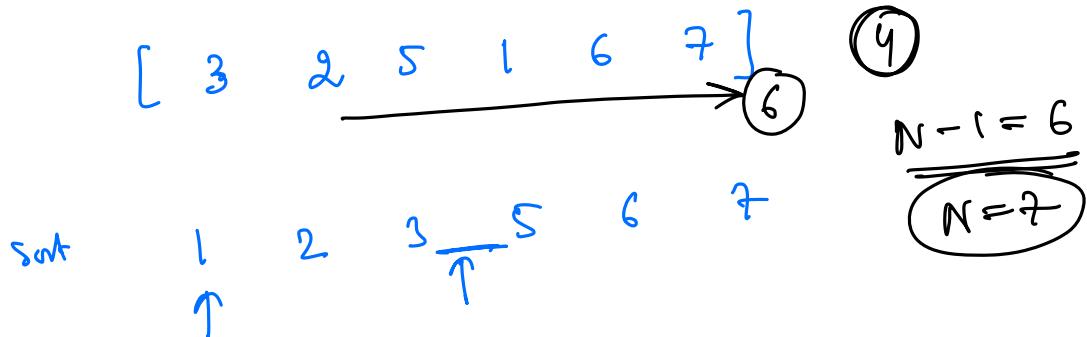
Q

Given an Array A of size  $(N-1)$ .

All the elements in the array are distinct.

$$1 \leq A[i] \leq N$$

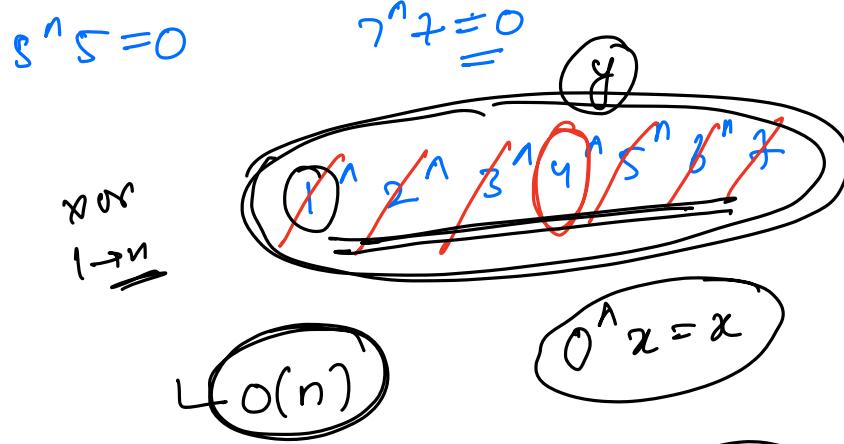
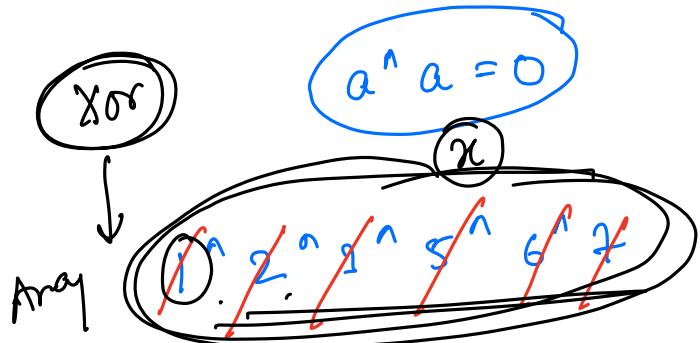
Find the missing element.



$$\text{sum} = \frac{\text{sum of } 1 \rightarrow N}{2} - O(1)$$

$$\text{sum of array} = 1 + 2 + 3 + 5 + 6 + 7$$

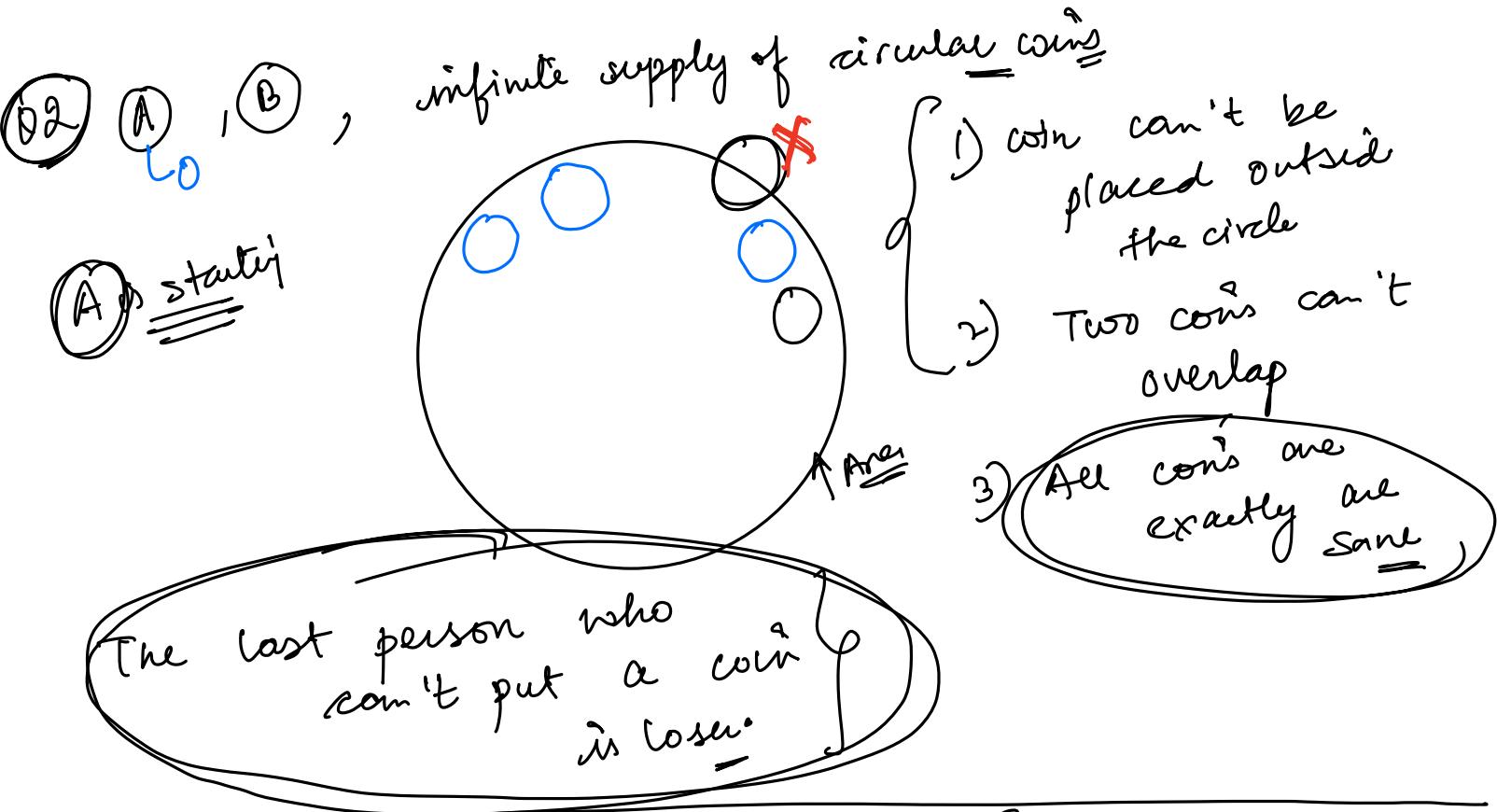
$$\text{sum} - \text{arraysum} = \text{missing element} \quad - O(n)$$



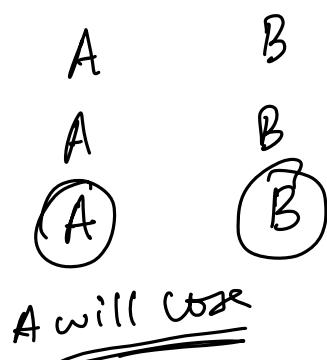
$$x=0$$
  
 $x^n y$   
 $\text{for } i=0 \rightarrow n$

$$\frac{n(n+1)}{2}$$

$y \leftarrow x^n = arr[i:j]$   
 for( $i=1 \rightarrow n$ )  
 $y^n = x^n$   
 mss =  $x^n y$

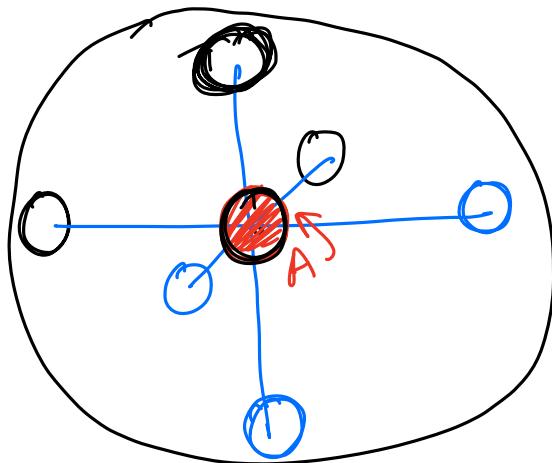


B can always person a similar move like A



Copy-cat

There is only  
one wheel  
can't be  
copied

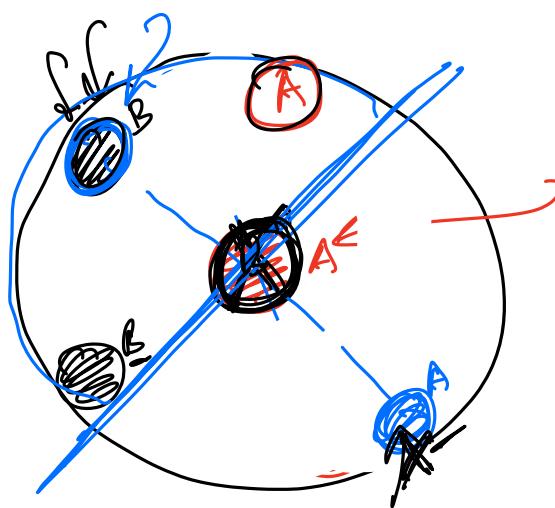


B will win

A - find a  
more  
B can't copy

place on  
center

A is  
optimal

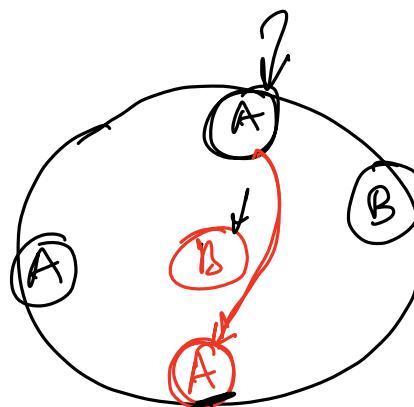


losing

optimal

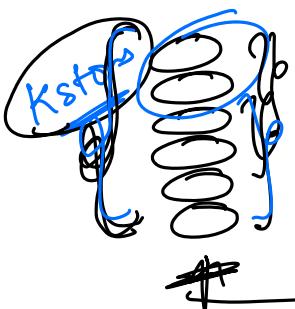
extra 'A'

jail-B



A is starting the game

Prime game



A, B

move'

any person - choose a pile  
& remove some stones

choose  
of  
pile  
&  
remove  
stones

initial  
stone

y  
remain  
stone

Coprime

$\text{gcd}(n, y) = 1$

(n,y)

HCF

greatest  
common  
div

last person  
who  
can't  
remove  
stones

y  
loser

# you can't  
empty  
the pile

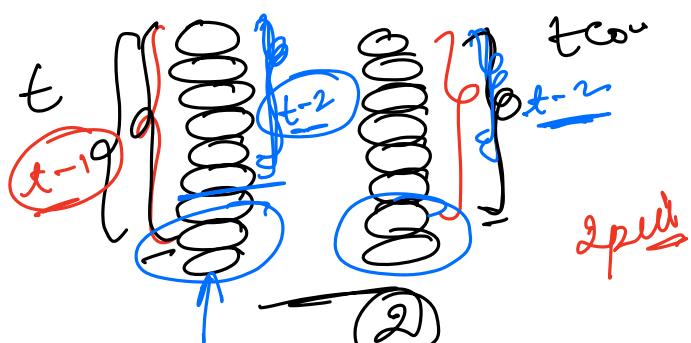
2  
9  
y  
8  
7

9 stone  
1  
8  
1  
3  
6  
3  
9 cd (11^2)

t stones  
t-1 coins

A

HCF  $(t, t-1)$  1

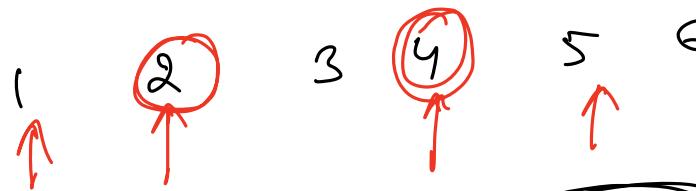


B ✓

symmetric

even no of piles =

if



Copy of  
New

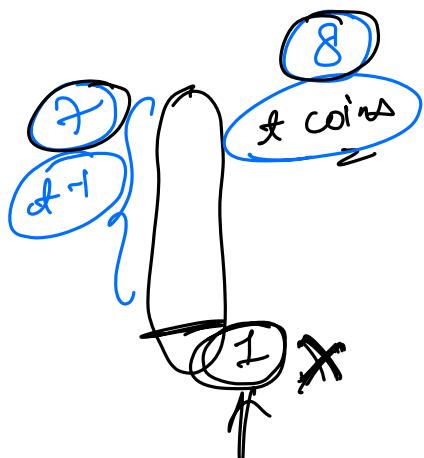
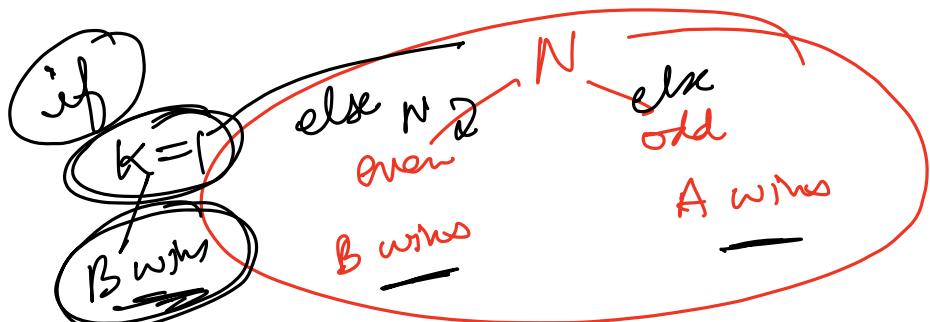
B

$N$  is even  
A will lose

$N$  - odd



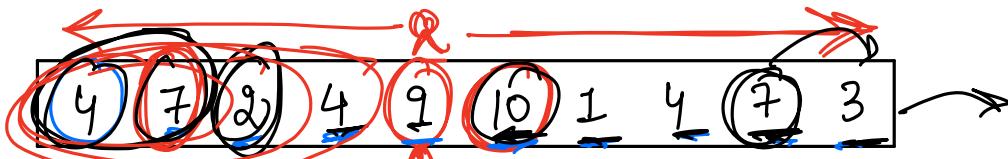
B  
Case



④

array with  $N$  integers. For every element, find the peak element on left & right side.

$N$

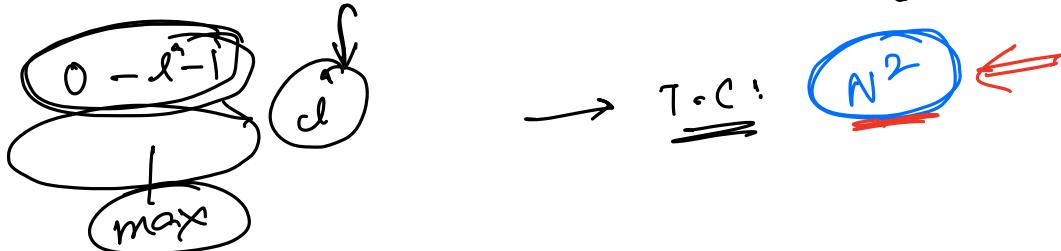


wft

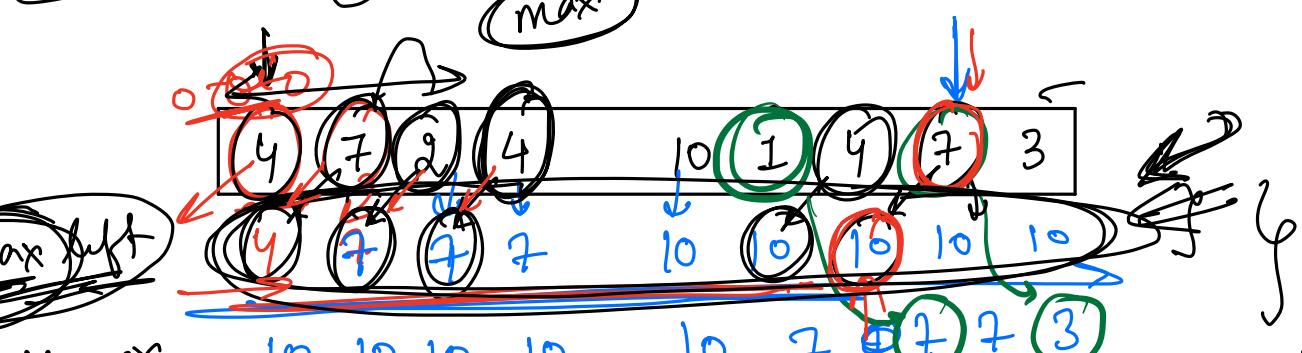
wgtr

B.F.

0 4 7 2 7 9 10 10 10 10  
10 10 10 10 10 7 7 7 3 0



$\sum_{0-l}^{x}$  sum of all  $0-l$   
 $\text{pf} \rightarrow$   
 $\text{pf}_{\text{max}} \rightarrow$  max of all  $0-l$



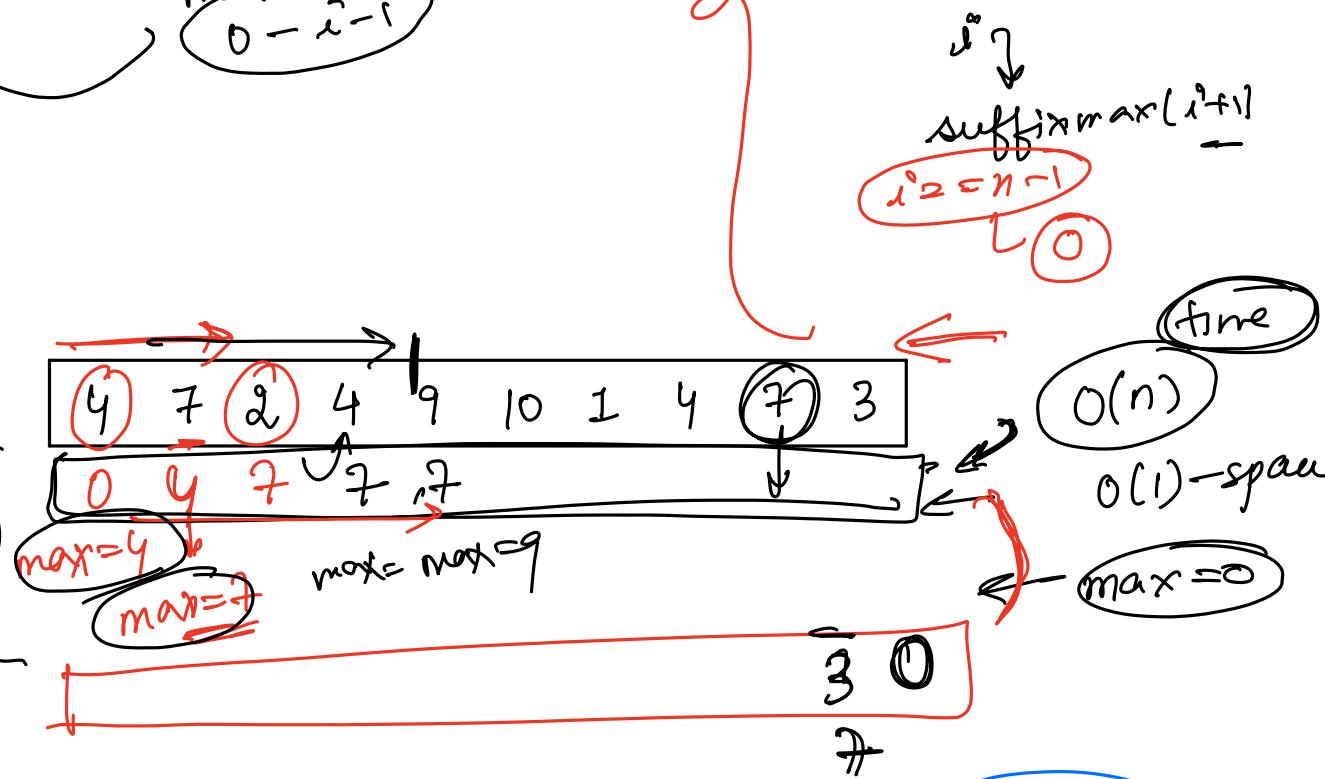
pt max left

pf buff max

$$pf_{\text{buff}}[i] = \max(pf[i-1], a[i])$$

$j \downarrow$   
 $pf_{\text{max}}[l-1]$

$j = \infty$   
0



```

int ans_left[10];
int max = 0;
for (i = 0; i < n; i++) {
    ans_left[i] = max;
    max = max(max(a[i]), max);
}
    
```

Annotations:

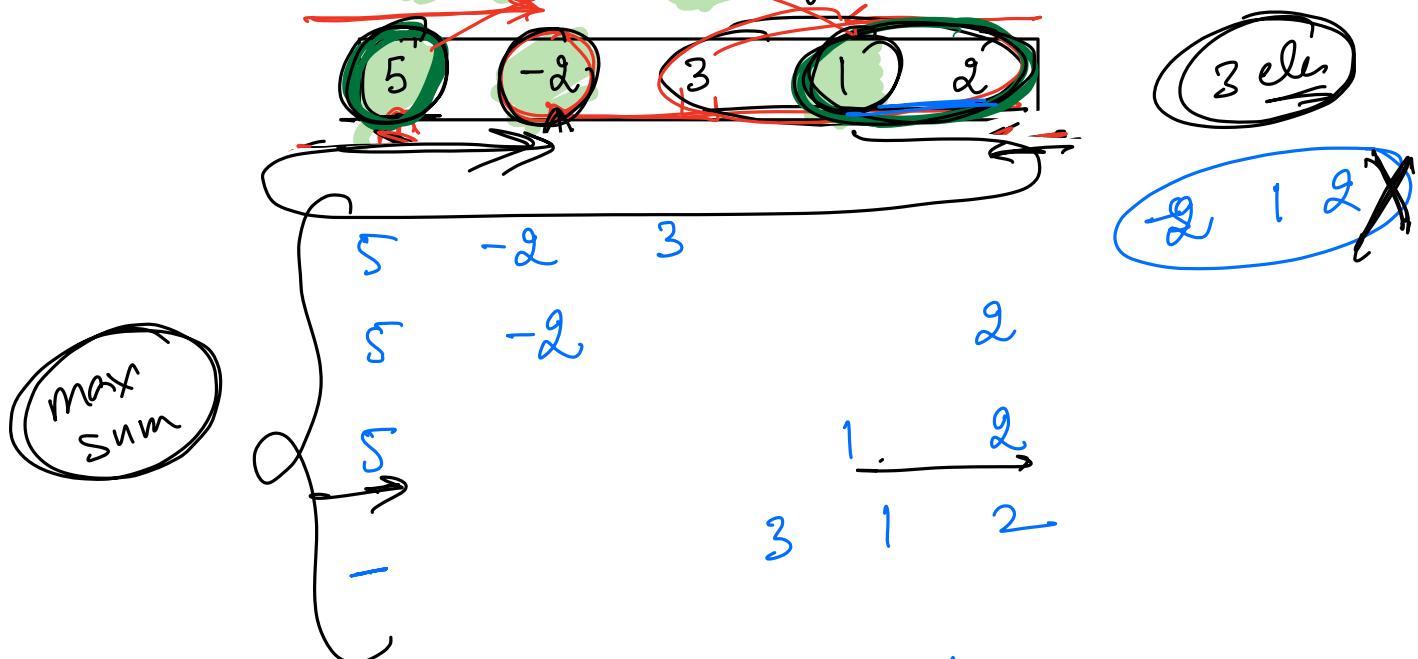
- 2 arrays
- output
- SC
- $O(n)$
- sol
- extra space used  
to solve  
the ques.

(S)

Q Pick from both sides

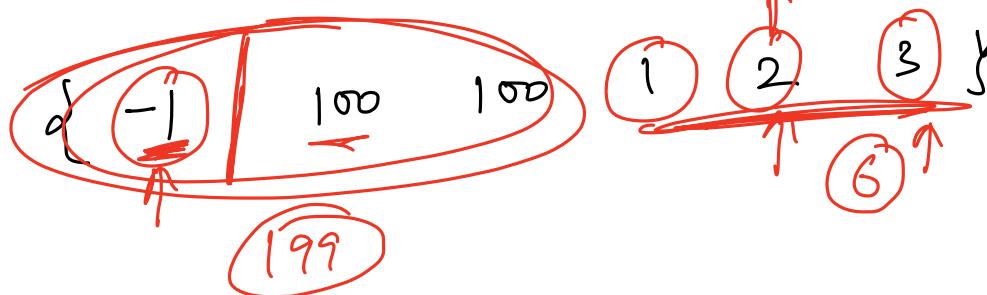
Given an integer array A of size N. You have to pick B elements. Elements can only be picked from either ends in a contiguous manner.

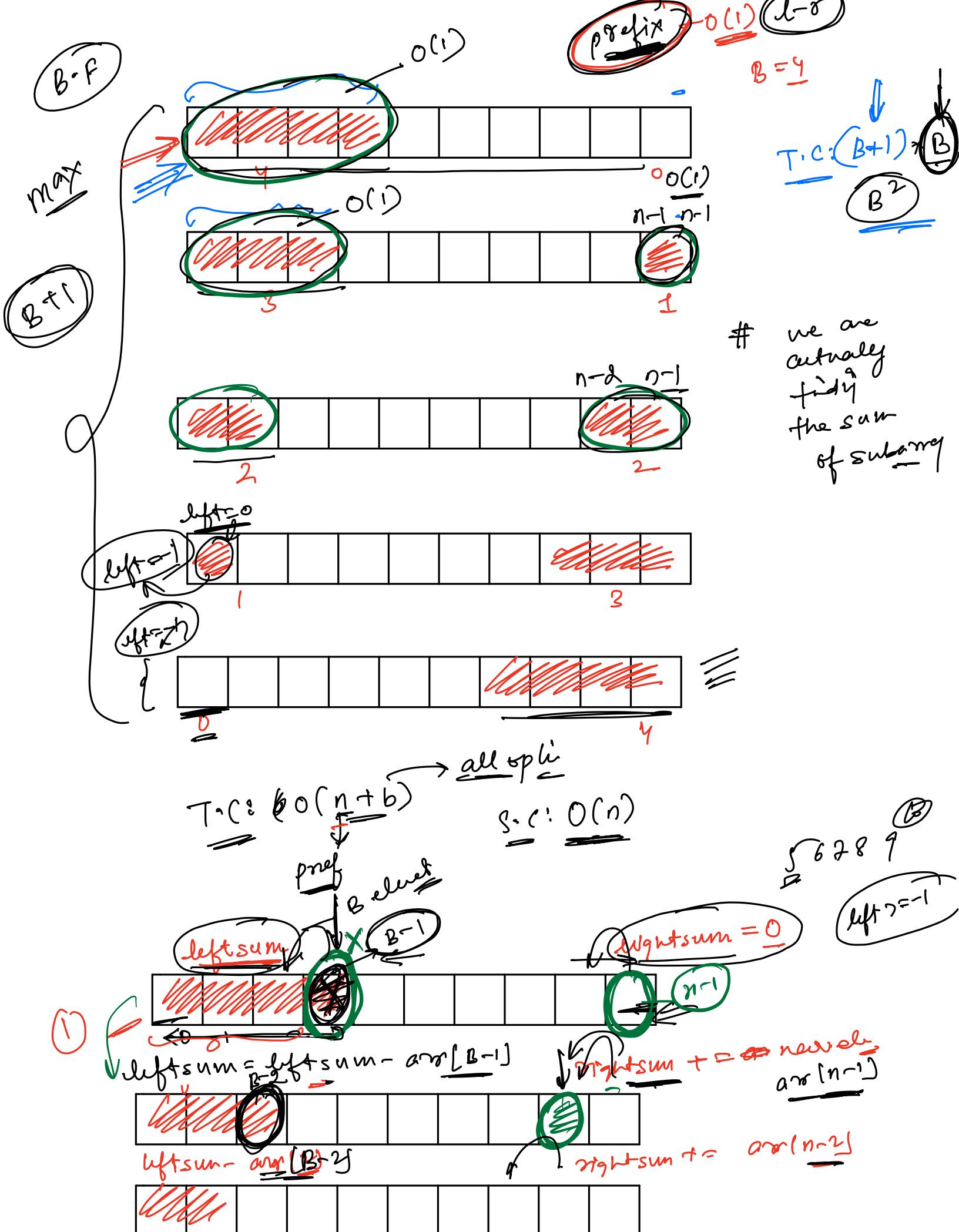
Find out the maximum sum you can obtain

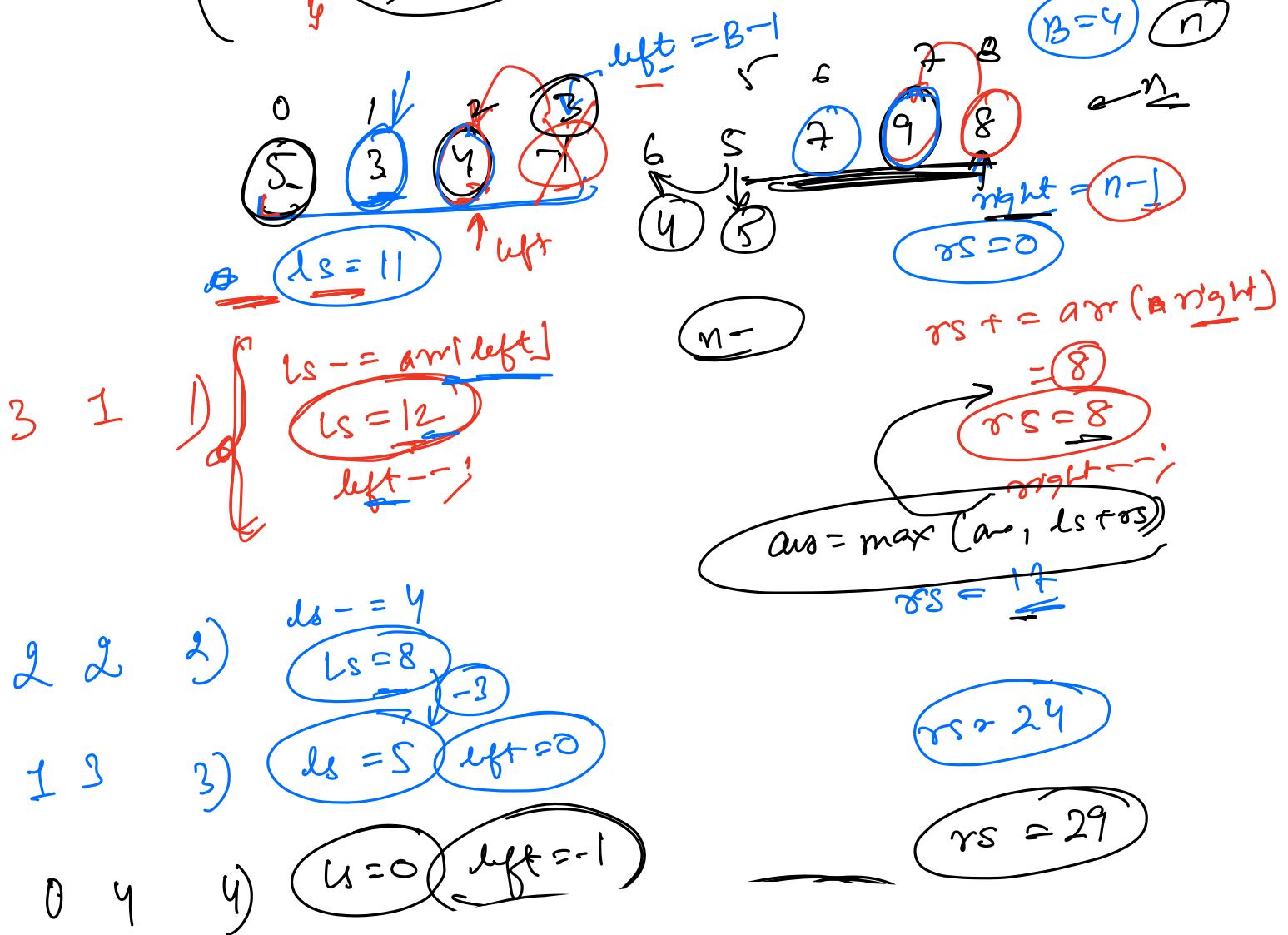
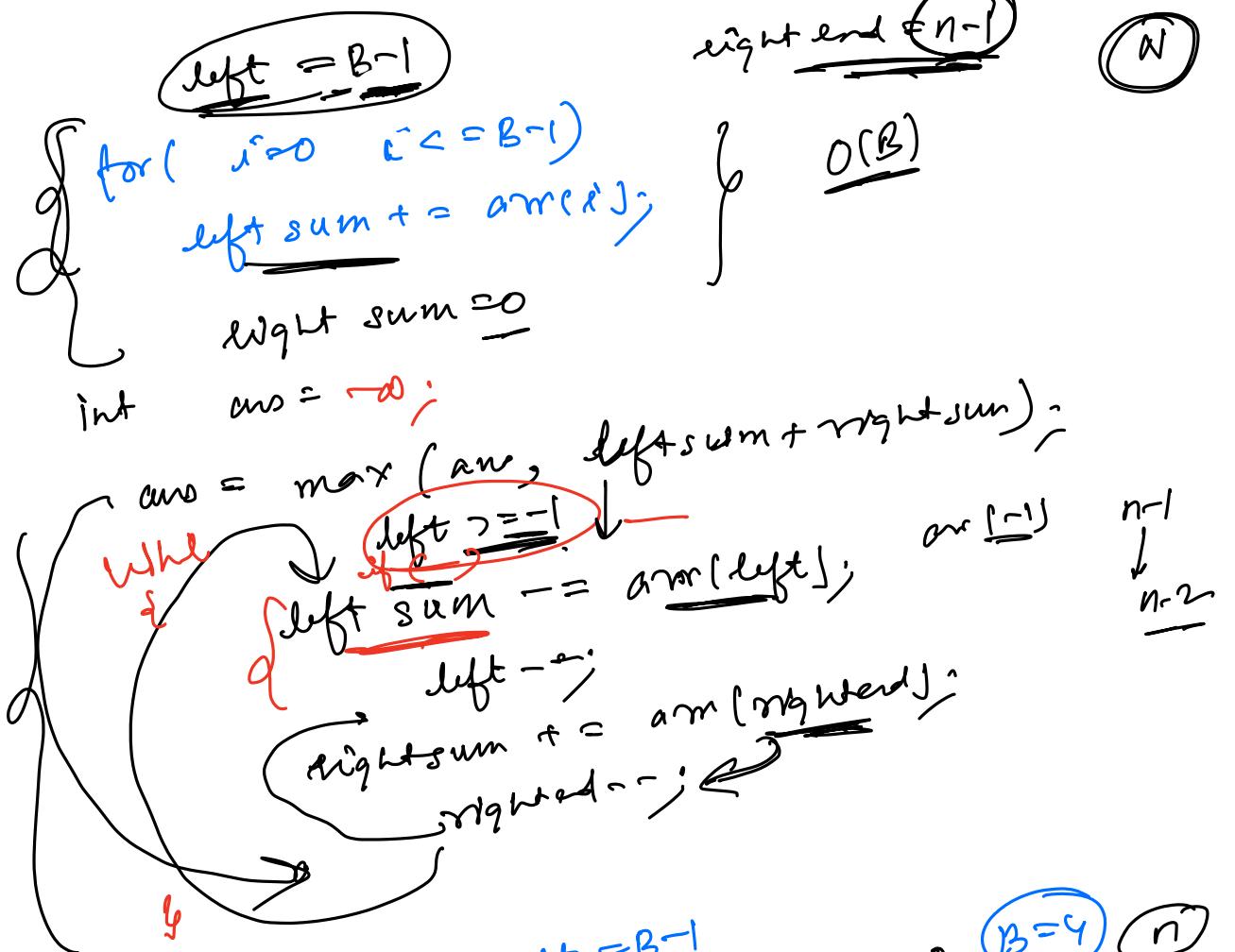


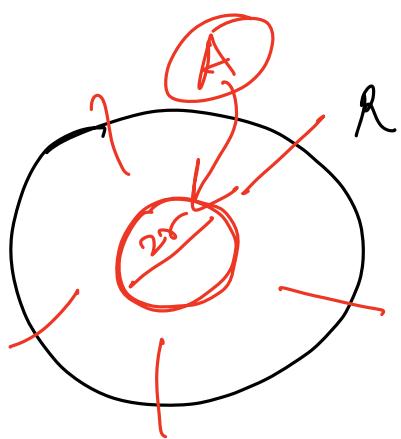
1) Sort array? *You'll lose the order*

2) two pointer - left end *right end*  
*greedy X* always choose max }  
*B = 3*









$$2r < R$$

$$\underline{2R < 6r}$$

$$\underline{R < 3r}$$

T<sub>a</sub>(s) O(B)

S<sub>a</sub>(s) O(C)

