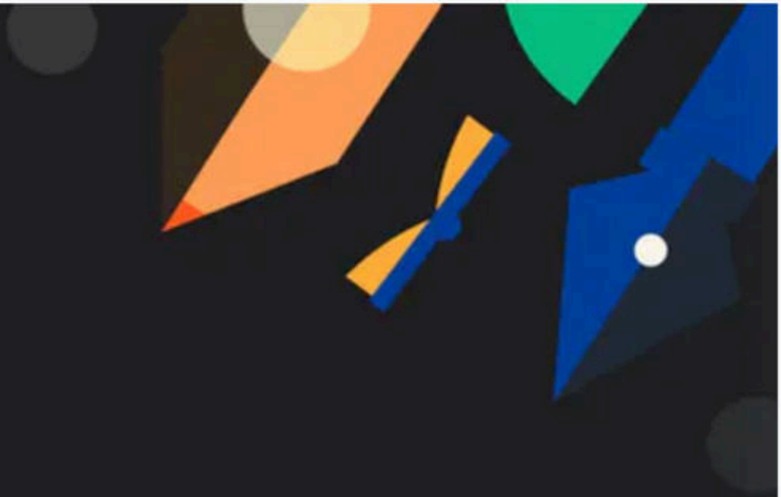
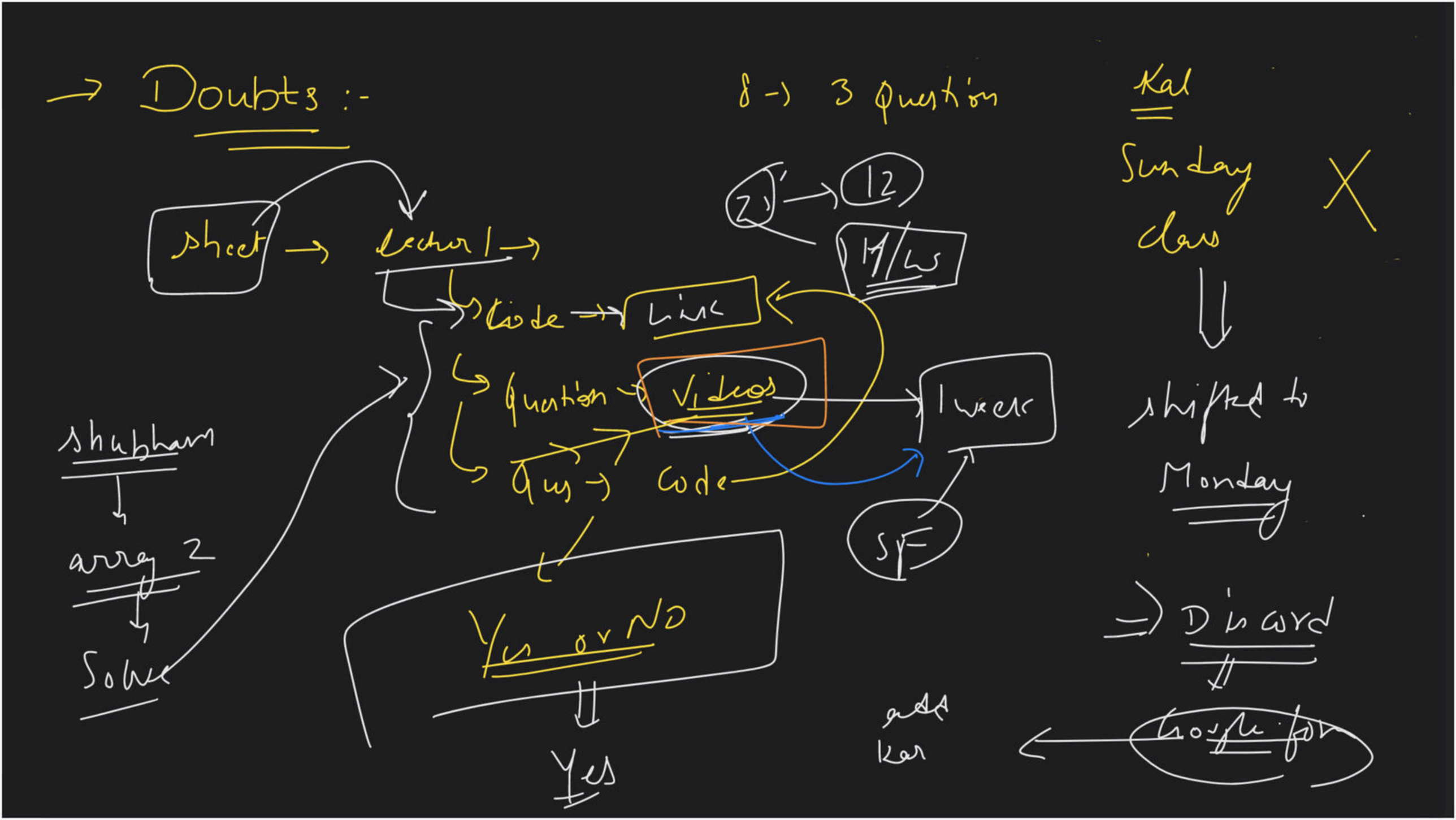


Doubt Clearing Session - Part III

Foundation Course on Data Structures & Algorithm - III

Love Babbar • Lesson 3 • May 2, 2022


→ Doubts :-
8 -> 3 Question
Kal
Sunday
class
X
shifted to
Monday
sheet →
Lecture →
Code → Link
Question → Videos
Ques → Code
21 → 12
M/Cs
Lecture
SF
Shubham
array 2
Solve
Yes or NO
Yes
⇒ Discard
Google for
add
kar

Array : 1D Array

→ Doubts

→ Min swaps: required to group <= K element together

K = 3

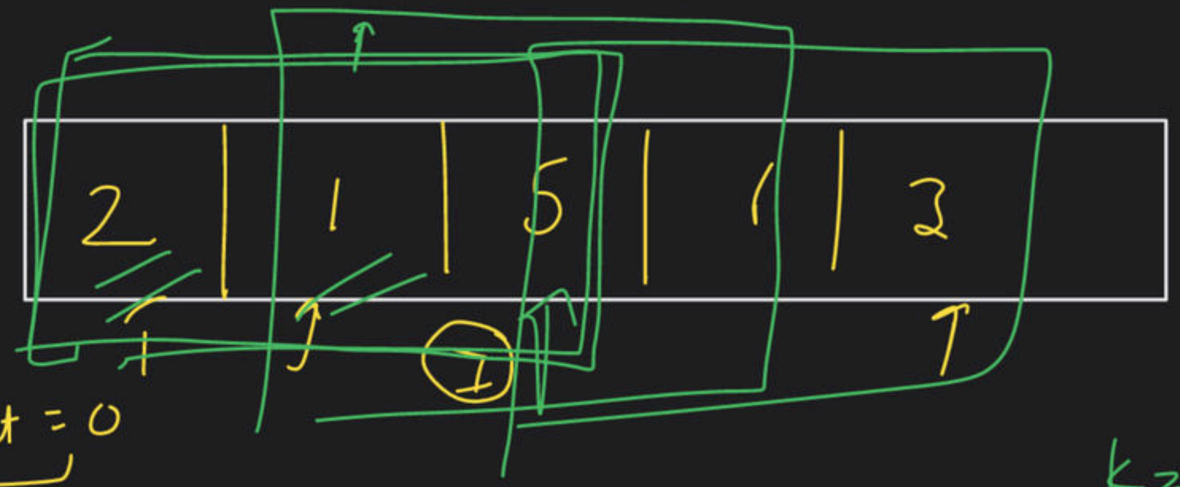
i/p { 2, 1, 5, 6, 3 }

→

→ { 2, 1, 3, 6, 5 }

swap = 1 ← ans

count of numbers <= k




```
for(i->0 < n)
{  if(arr(i) <= k)
      count ++;
}
```

count = 3

gaada bad = 1

k=3

2 <= k =

1 <= k

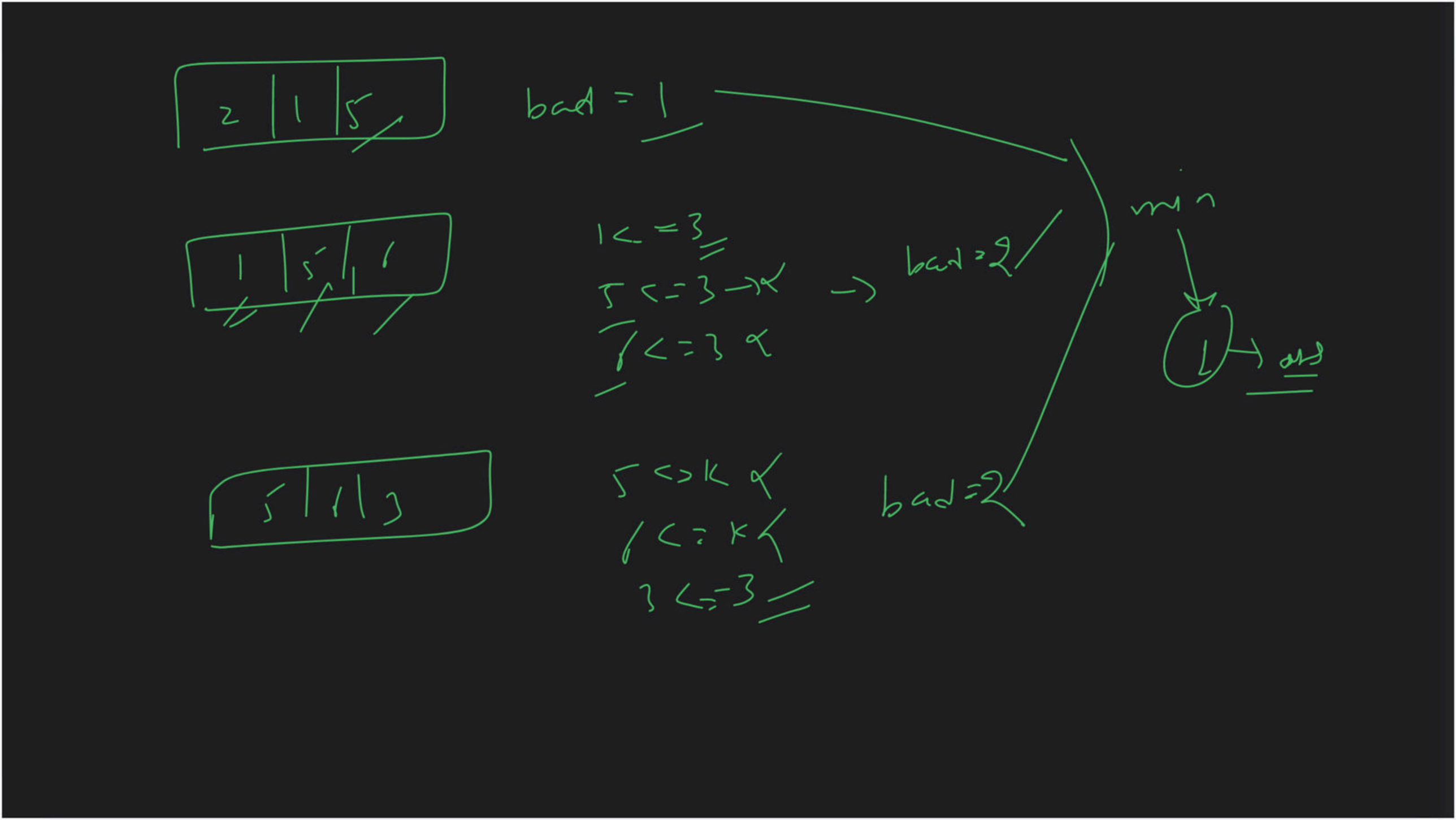
5 <= k !

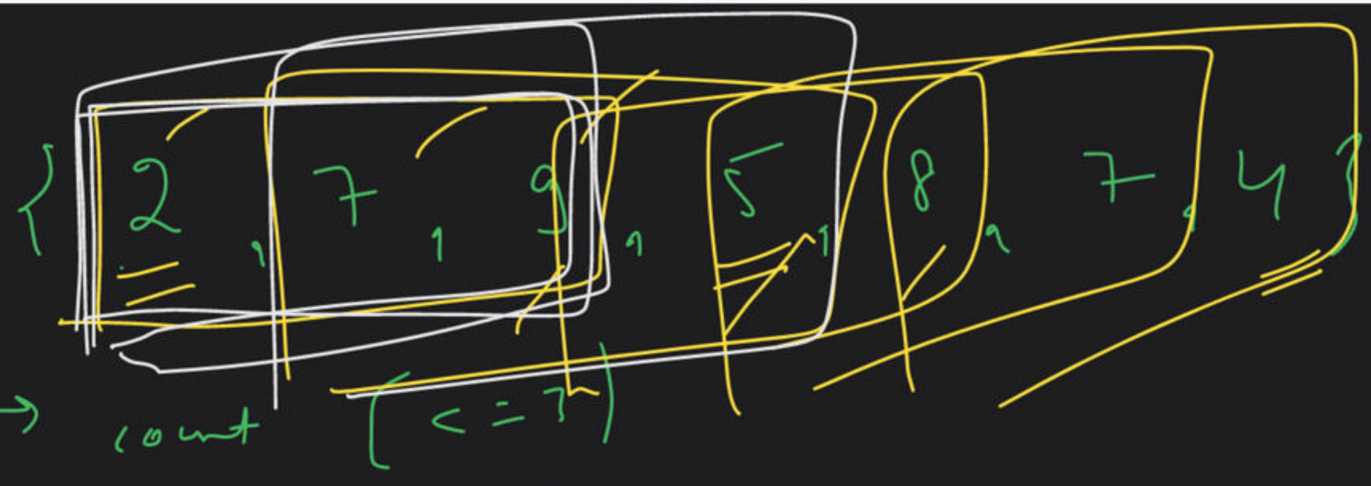
<= F

Intuition -> ?

approach

<= k


2 | 1 | 5
bad = 1
1 | 5 | 1
1 <= 3
5 <= 3 -> x
1 <= 3 ✓
-> bad = 2
5 | 1 | 3
5 <> k ✓
1 <= k ✓
3 <= 3
bad = 2
min
1
ans



k = 5

count = 3

window size

2 | 7 | 9 → bad → 2

7 | 9 | 5 → bad → 2

5 | 5 | 8 → bad → 2

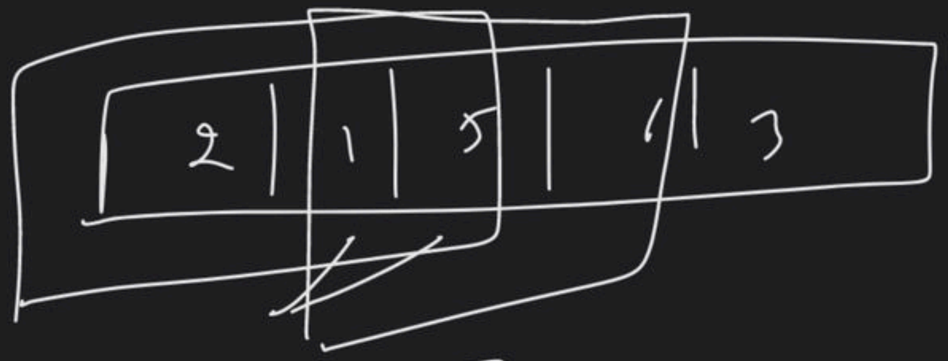
> k → > 5

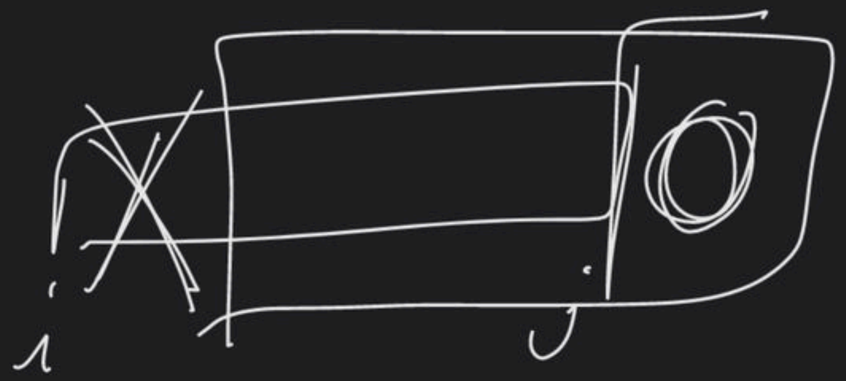
5 | 9 | 7 → bad = 2

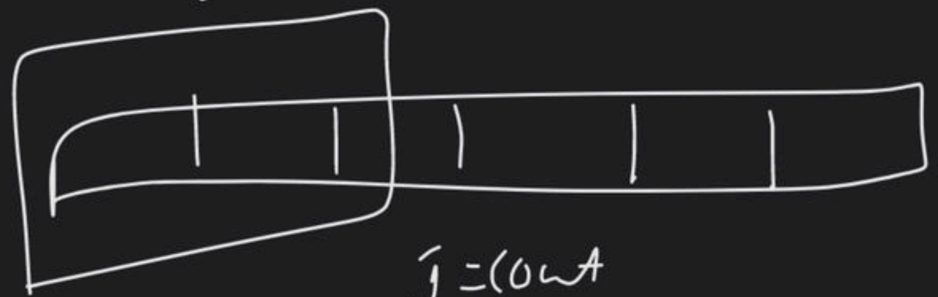
9 | 7 | 4 → 2

ans = 2

SW







i=0

j=count

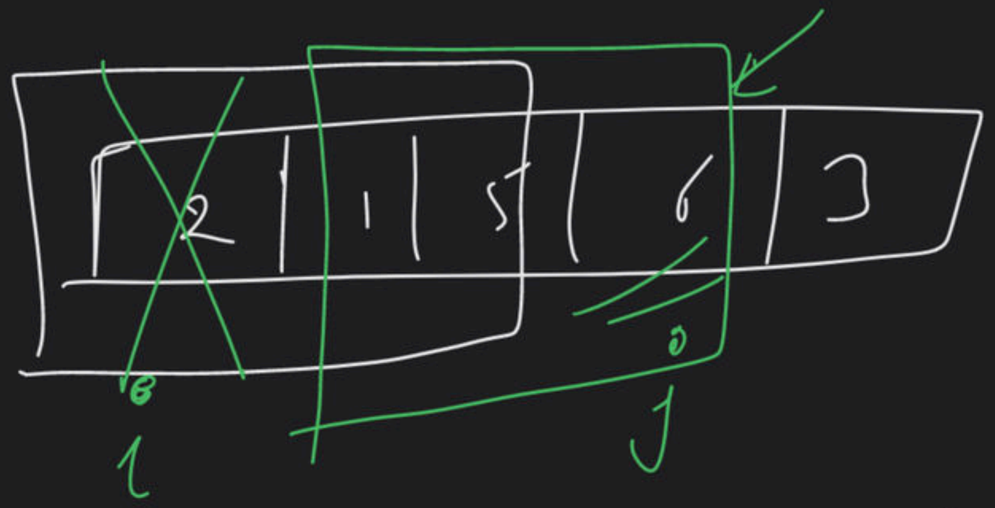
i++ j++

if (arr[i] > k)

bad -= 1

if (arr[j] > k)

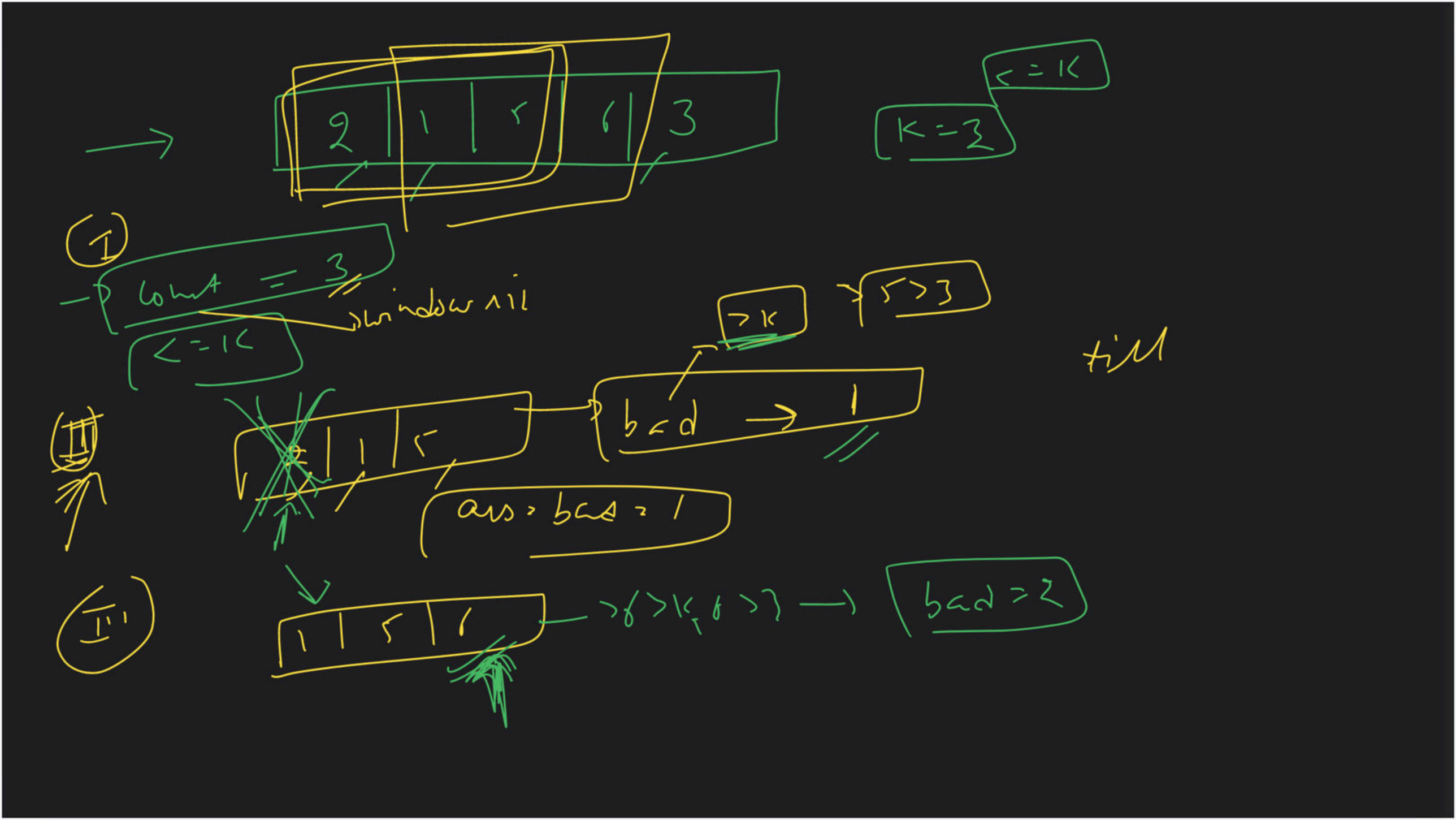
bad++

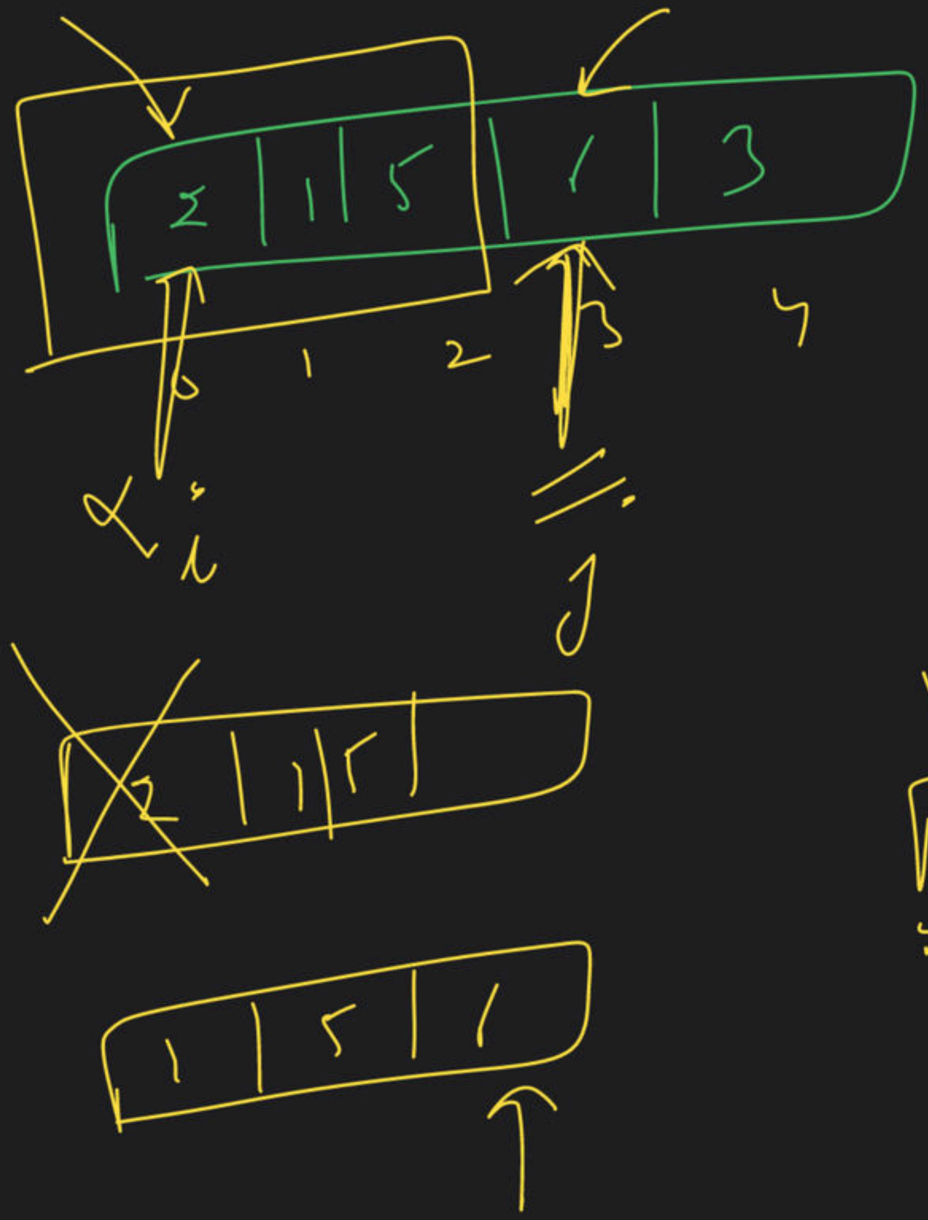



```
if(arr[i] > k)
    bad--
```

```
if (arr[j] > k)
    bad++;
```

```
ans = min(ans, bad)
}
```


2
1
r
6
3
<=k
k=3
(I)
count = 3
window nil
<=k
>k
>3
till
bad → 1
ans = bad = 1
(II)
1
r
r
>3 >k >3
bad=3



(iii)

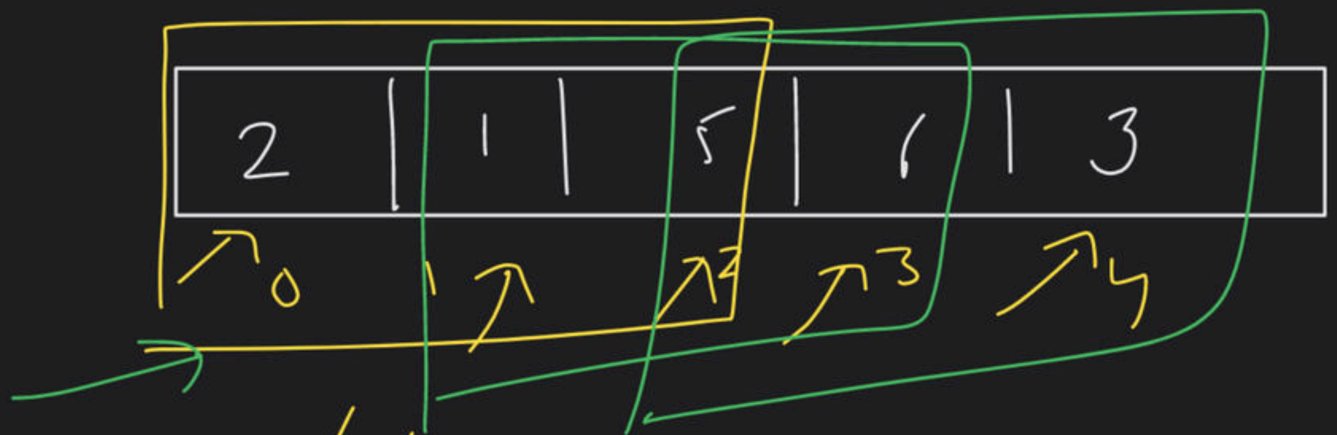
$i = 0$

$j = \text{out} = 3$

T.C $= O(n) + O(n) + O(1)$

$= O(n)$

interview ke



window size -> hi - lo + 1 <= k

k = 3

```
if (arr[i] > k)
    bad--;
```

(I) count = 0 / 2

3

2 <= 3 -> T

1 <= 3 -> T

5 <= 3 -> F

6 <= 3 -> F

3 <= 3 -> T

(II)

bad = 0

2 > k -> F

1 > k -> F

j > 3 -> T

```
if (arr[j] > k)
    bad++;
```

bad = 0 / 1

bad = 1

for(i=0, j=3 ; j<n ; i++, j++)
{
}

bad → no > k

i → element to remove

j → ——— Add

i 0 1 -2

j 3 4 5

result gen

min

2 > 3

2 > k → F

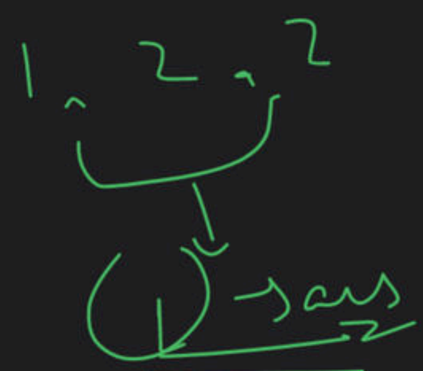
6 > 3 → T

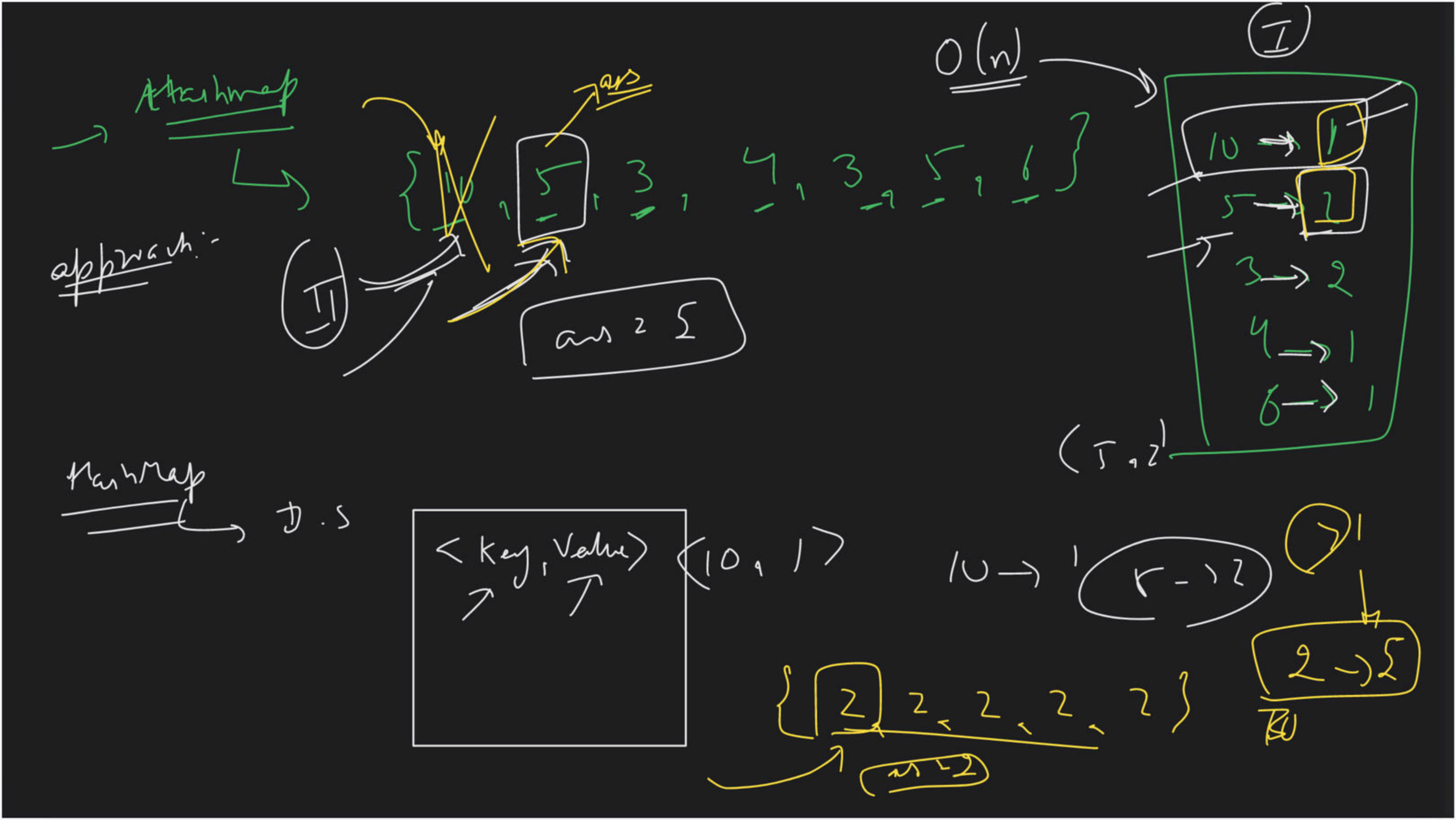
bad = 2

1 > 3 → F

3 > 3 → F

→ bad = 2




Attachment
approach:-
ans
ans = 5
O(n)
T
10 → 1
5 → 2
3 → 2
4 → 1
6 → 1
(5, 2)
HashMap
D.S
<key, Value>
<10, 1>
10 → 1
1 → 2
2 → 5
2, 2, 2, 2, 2
ans = 2

→ Algo:-

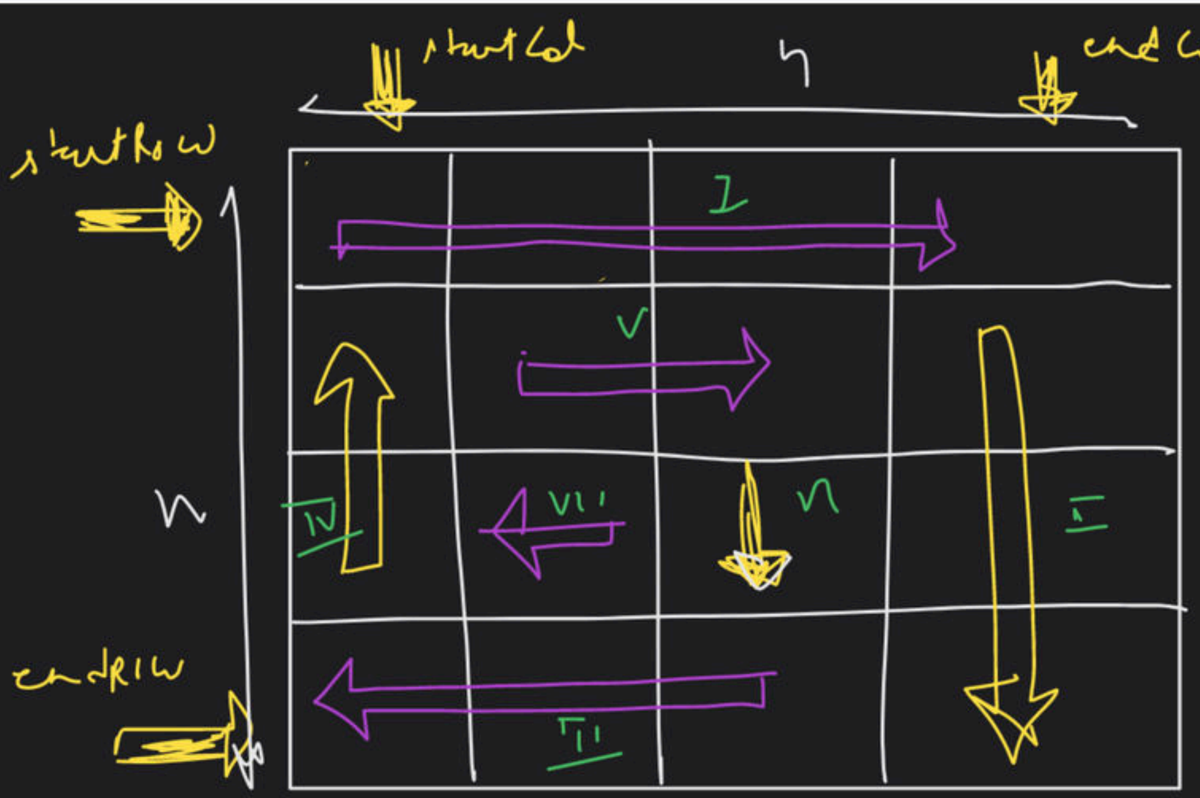
↳ (I) → mapping → <num → count>

```
unordered_map<int, int> mapi;
```

↳ (II) → traverse array

↳ check each element-

```
if (element_count > 1)
    cout << num;
```



issue-?

Go

2 approach

count = 0

while (count < total)

Total = n × n

= n^2

[10, 1, 2, 3, 4]

map<int, int> m;

m[10]++

m[10] → 0 1

<10, 1>

→ min platform problem

arrival → [a, b, c, d]

departure → [e, f, g, h]

minimum platform ?

approach :

1) sort arrn / dep

arr → 9 am 11 am

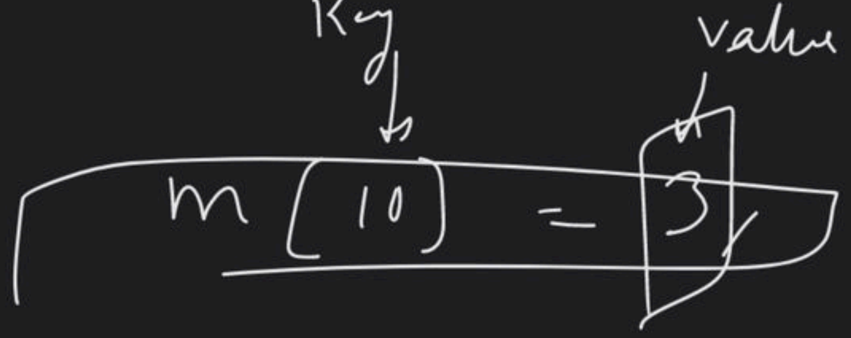
dep → 1 pm 2 pm

2)

map <int, int> m

key

value




```
if (m[10] > 1)
    cout <<
```

access

m[10]

Adobe / MS / Amazon

arr → { 9:00, 9:40, 9:50, 11, 15, 18 }

dep → { 9:10, 12:00, 11:20, 11:30, 15:00, 20:00 }

(BMK

(1)

arr → { 9:00, 5:40, 9:50, 11, 15, 18 }

dep → { 9:10, 11:20, 11:30, 12:00, 15:00, 20:00 }

(2) platforms = 1 0 1 2 3 2 1 0 1 2 1 0

10

max → ans = 3

الحل الصحيح

int sum = 0, int maxi = INT_MIN

for (i -> 0 -> n)

{

I → sum = sum + arr(i)

II → maxi = max(maxi, sum)

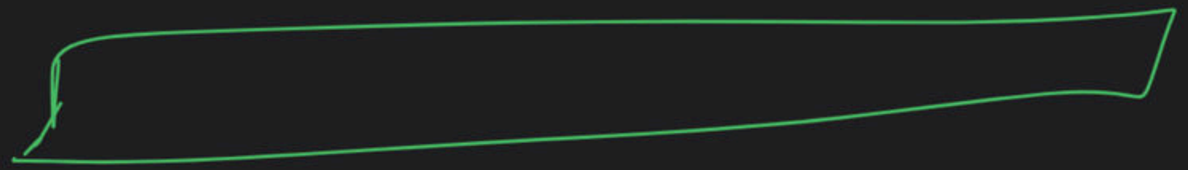
III → if (sum < 0)

sum = 0

}

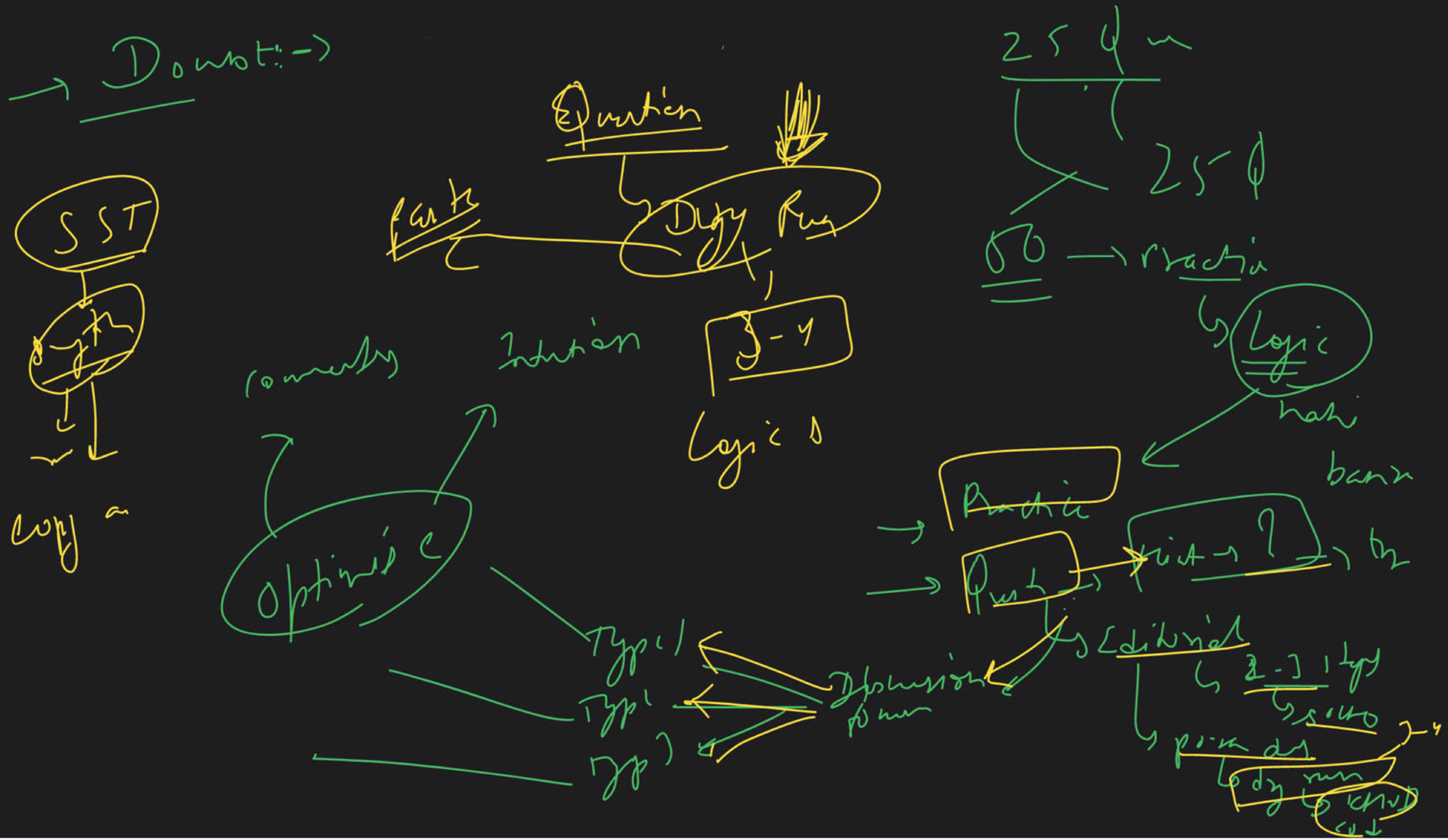
→ Kadane's

Kode



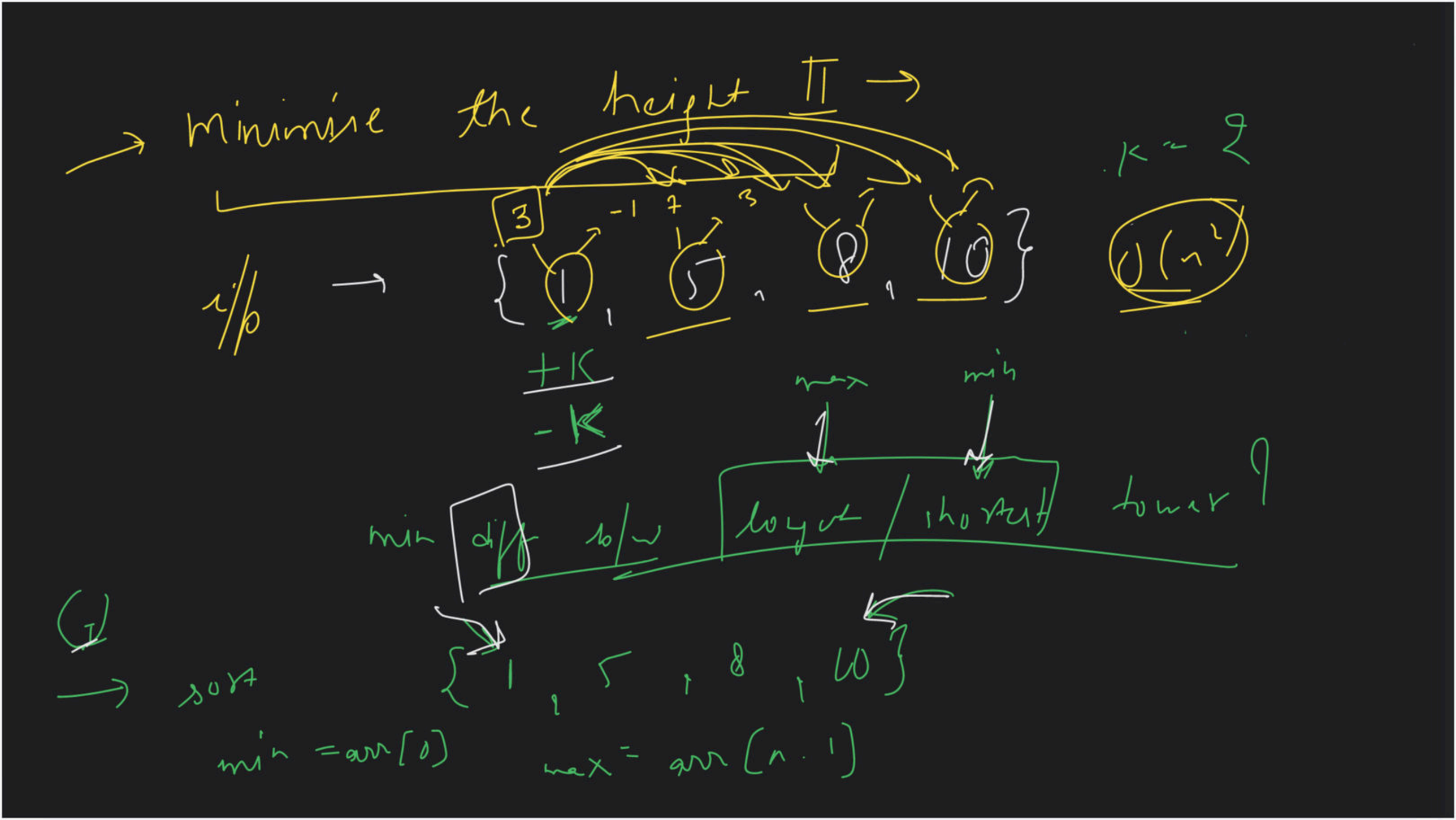
```
for (i      )                    brute for n
  {
    for(j - i - n)
      {

      }
  }
```



Doubt:-
Question
Facts
Dry Run
SST
Intuition
Logic
25 Q
Optimise
Type 1
Type 2
Type 3
Editorial

Late :→


Minimise the height
3
-1
7
3
1
5
8
10
K = 2
O(n²)
i/p
+K
-K
max
min
min diff
logic / shortest tower ?
sort
{ 1 , 5 , 8 , 10 }
min = arr[0]
max = arr[n-1]

k = 2

{ 1 , 5 , 8 , 10 }

(1) Tower height → +k, -k

(1) min diff → Largest − smallest

| a | b | c | d |

shrink → a + k

Tower → d + k

a - k , d - k

= d + k − a − k

= d − a

= L − S

= d − k − (a − k)

= d − k − a + k

= d − a

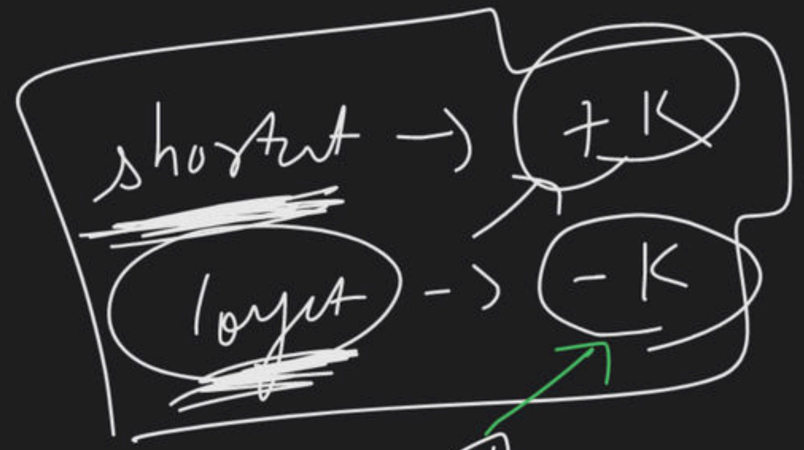
+k , +k

+k , -k

-k , -k

-k , +k

a - k a d + k



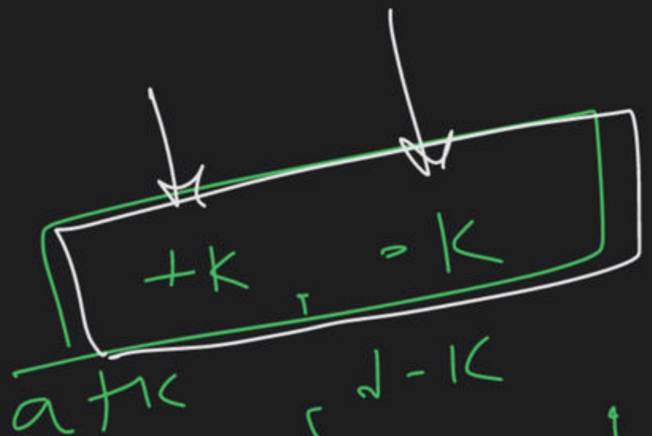
Observation

(b ≠ k

= L - S = d + k - (a - k)

= d + k - a + 2k

= d - a + 2k



L - S ⟹ d - k - (a + k)

= d - k - a - k

= d - a - 2k

(1) Min

shortest { 1, 5, 8, 10 }

K = 2

1 + 2 = 3

10 − 2 = 8

→ 5

+K, −K → max, min → ans

{ 1, 5, 8, 10 }

arr

arr(n−1)

mini = min [arr(0) + K, arr(i) − K];

maxi = max [arr(n−1) − K, arr(i) + K]

Code :-

(1) sort (arr, arr + n); $n \log n$

(2) mini = arr[0];
maxi = arr[n-1];
ans = maxi - mini;

(3) for (i -> 1 -> n) → update

{
mini = min [arr[0] + K , arr[i] - K]

maxi = max [arr[n-1] - K , arr[i-1] + K]

ans = min (maxi - mini, ans);
}

return ans;

→ $O(n \log n)$

30 min

int mx = -10;

int val = abs(n);

5 examples

DRY RUN

copy

clear

paste

run