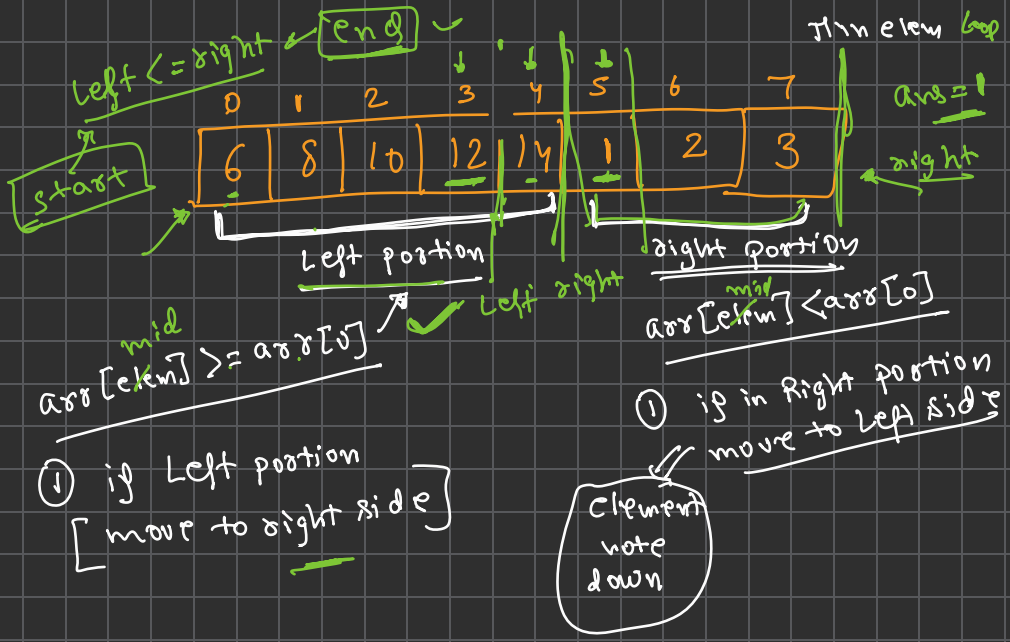
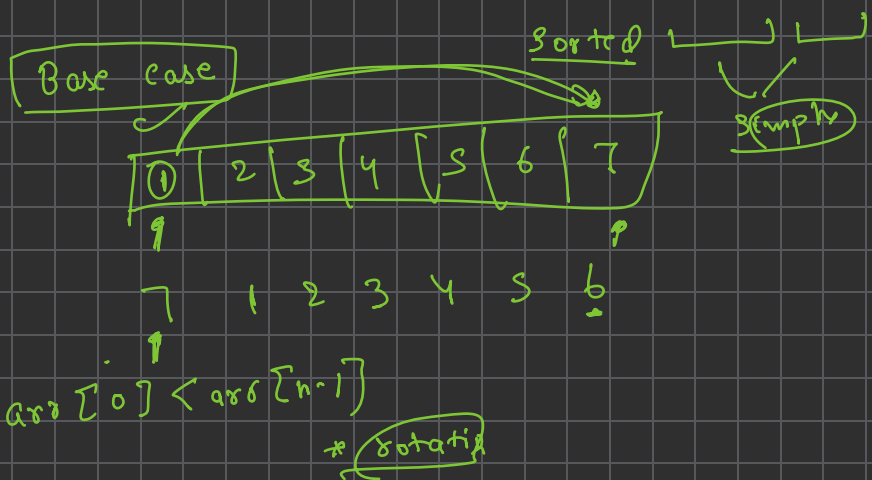
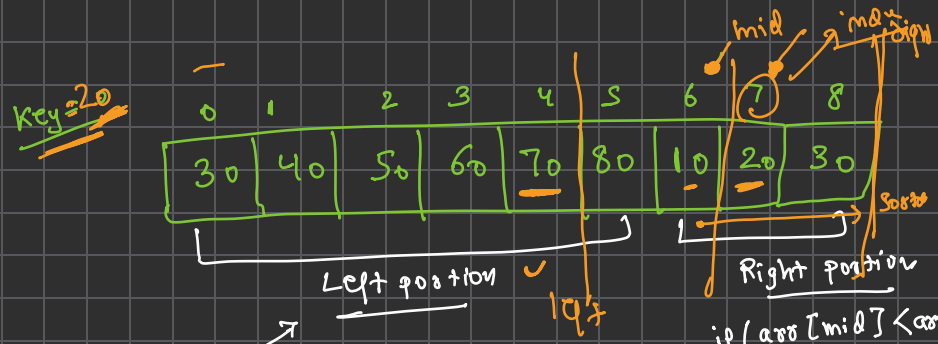
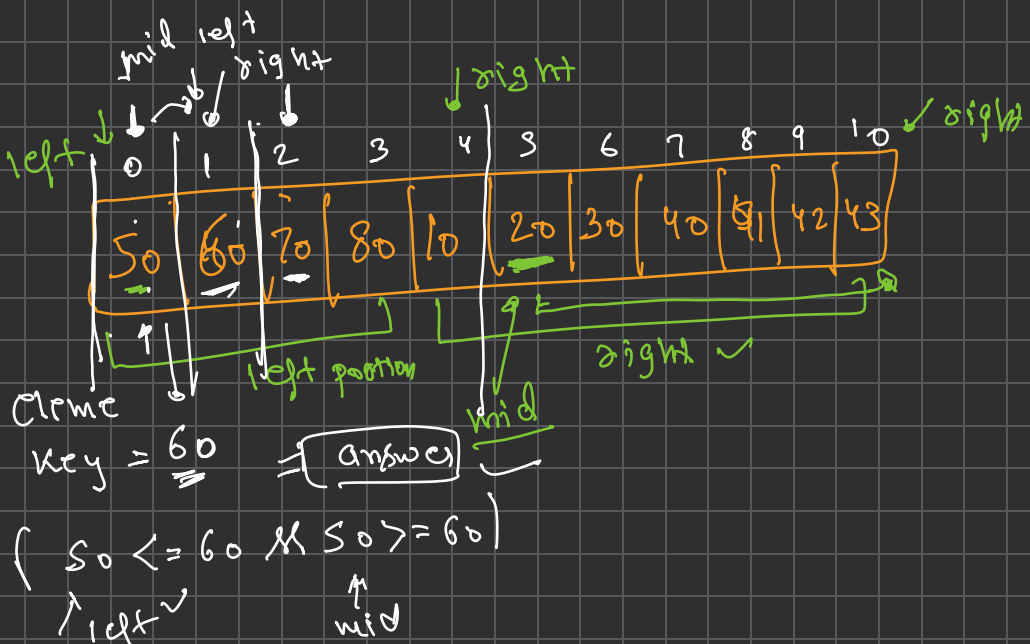

Min elem
left <= right
end
start
0 1 2 3 4 5 6 7
6 8 10 12 14 1 2 3
ans = 1
right
Left portion
right portion
left right
mid
arr[elem] >= arr[0]
arr[elem] < arr[0]
① if Left portion
[move to right side]
① if in Right portion
move to Left side
Element note down

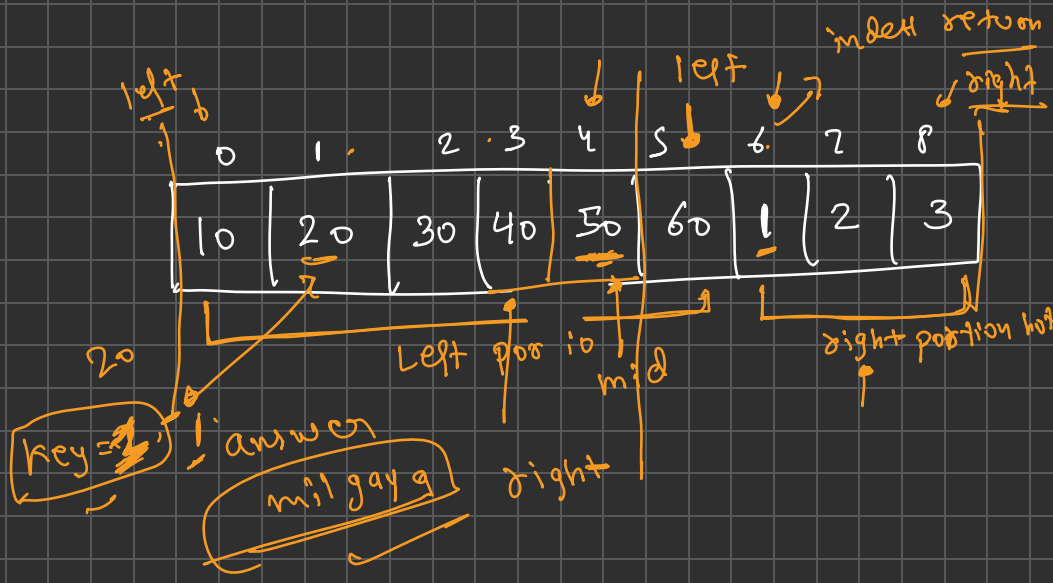

Base case
Sorted
Simple
0 2 3 4 5 6 7
7 1 2 3 4 5 6
arr[0] < arr[n-1]
* Rotation

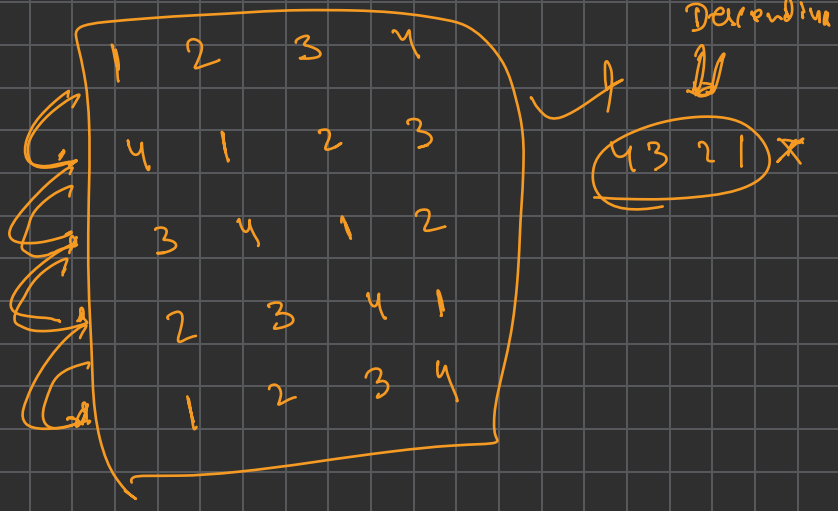


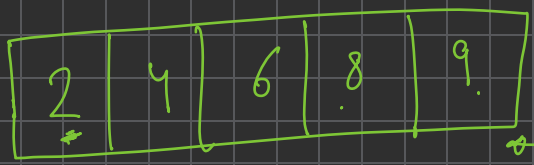

```
if (arr[mid] >= arr[0])
{
  if (key >= arr[left] &&
      key < arr[mid])
  {
    right = mid - 1;
  }
  else
  {
    left = mid + 1;
  }
}
```

```
if (arr[mid] < arr[0])
{
  if (key > arr[mid]
  && key <= arr[right])
  {
    left = mid + 1;
  }
  else
  {
    right = mid - 1;
  }
}
```


mid
left
right
right
left
right
0 1 2 3 4 5 6 7 8 9 10
50 60 70 80 10 20 30 40 41 42 43
left position
right
Clrmc
key = 60
answer
mid
(50 <= 60 && 50 >= 60)
left
mid


left
left
index return
right
0 1 2 3 4 5 6 7 8
10 20 30 40 50 60 1 2 3
20
Left pointer
mid
right position hua
key = 2
answer
mil gaya
right


1 2 3 4
4 1 2 3
3 4 1 2
2 3 4 1
1 2 3 4
Descending
4 3 2 1



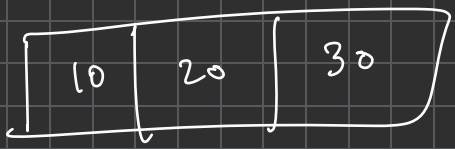
K=8 5

sorted

5 pos.

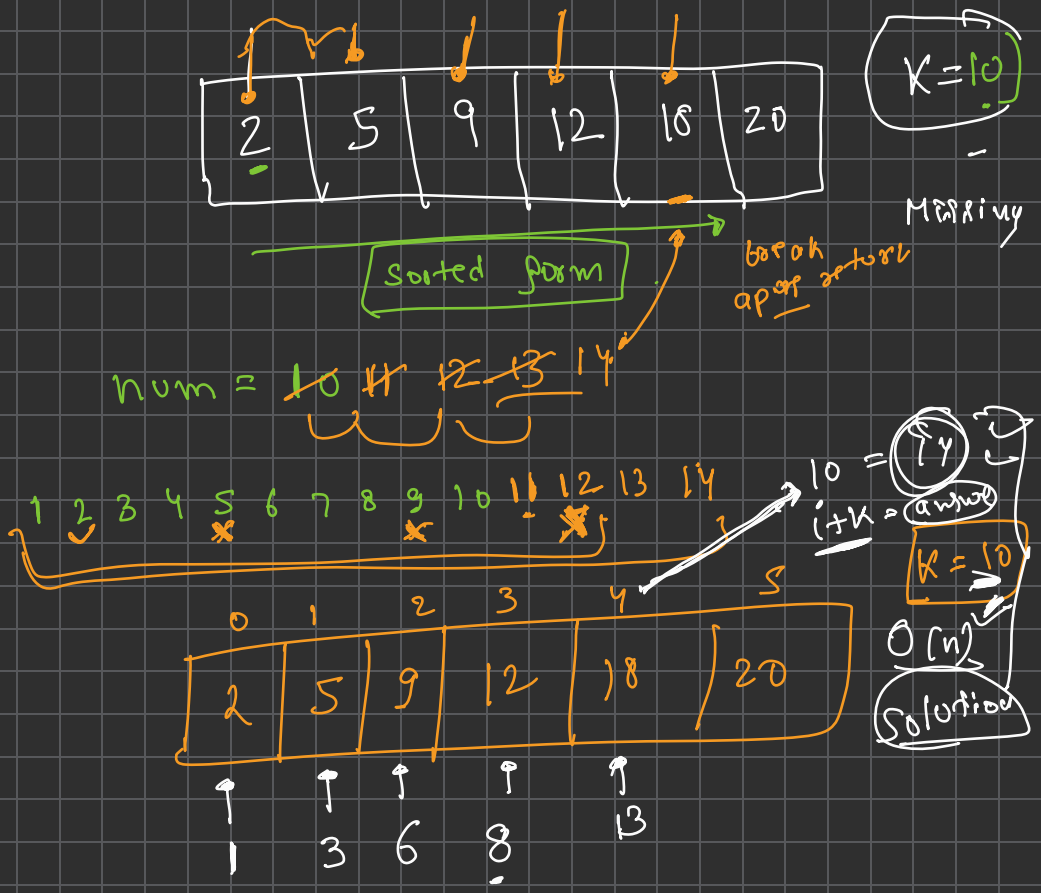
1 2 3 4 5 6 7 8 9 10

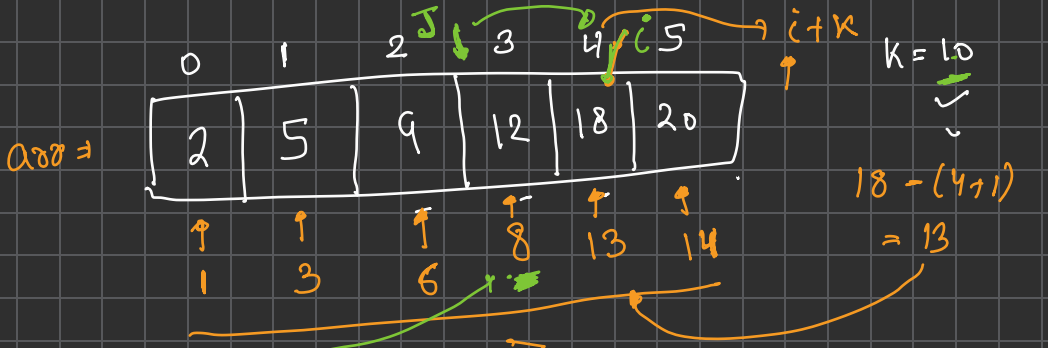
K=15



1 2 3 4 5 6 7 8 9 10

11 12 13 14 15 16


2
5
9
12
10
20
K=10
Missing
Sorted form
break
apply
return
num = 10 11 12 13 14
1 2 3 4 5 6 7 8 9 10 11 12 13 14
10 = 14
i+K = answer
K=10
O(n)
Solution
0 1 2 3 4 5
2 5 9 12 18 20
1 3 6 8 13



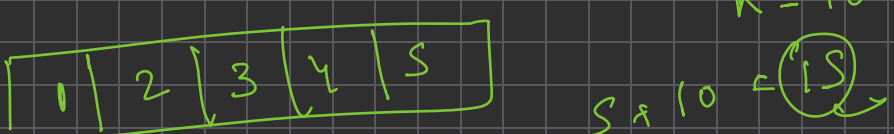
[total missing = arr[i] - (i+1)]

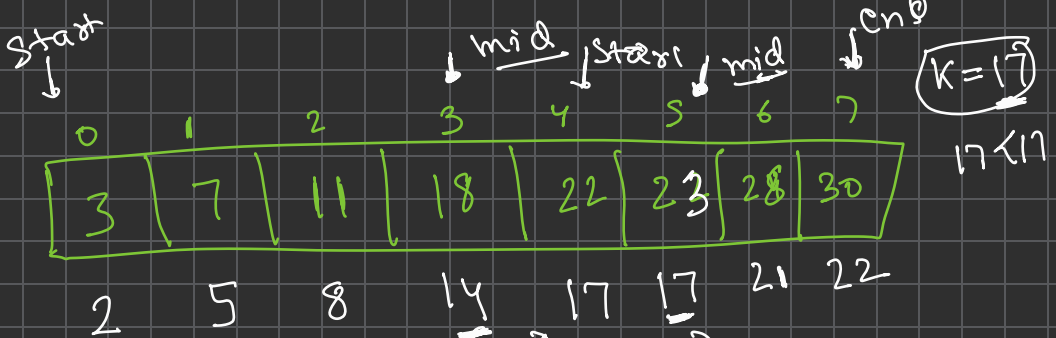
arr[j] + (10-8)

arr[j] + (K - [arr[j] - (j+1)])

K + (j+1)

[i+K]



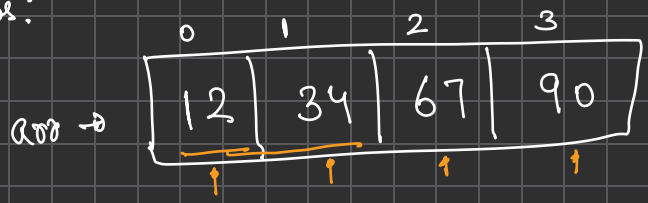



```
if( arr[mid] - (mid+1) < k )
    start = mid+1;

else {
    ans = mid;
    right = mid-1;
}
```

① → Every student, should rec at 1 book

② → Contigous:



K = 2 → number

I st

12 (12)

12 34 46

12 34 67 113

II nd

34 67 90 191

67 90 157

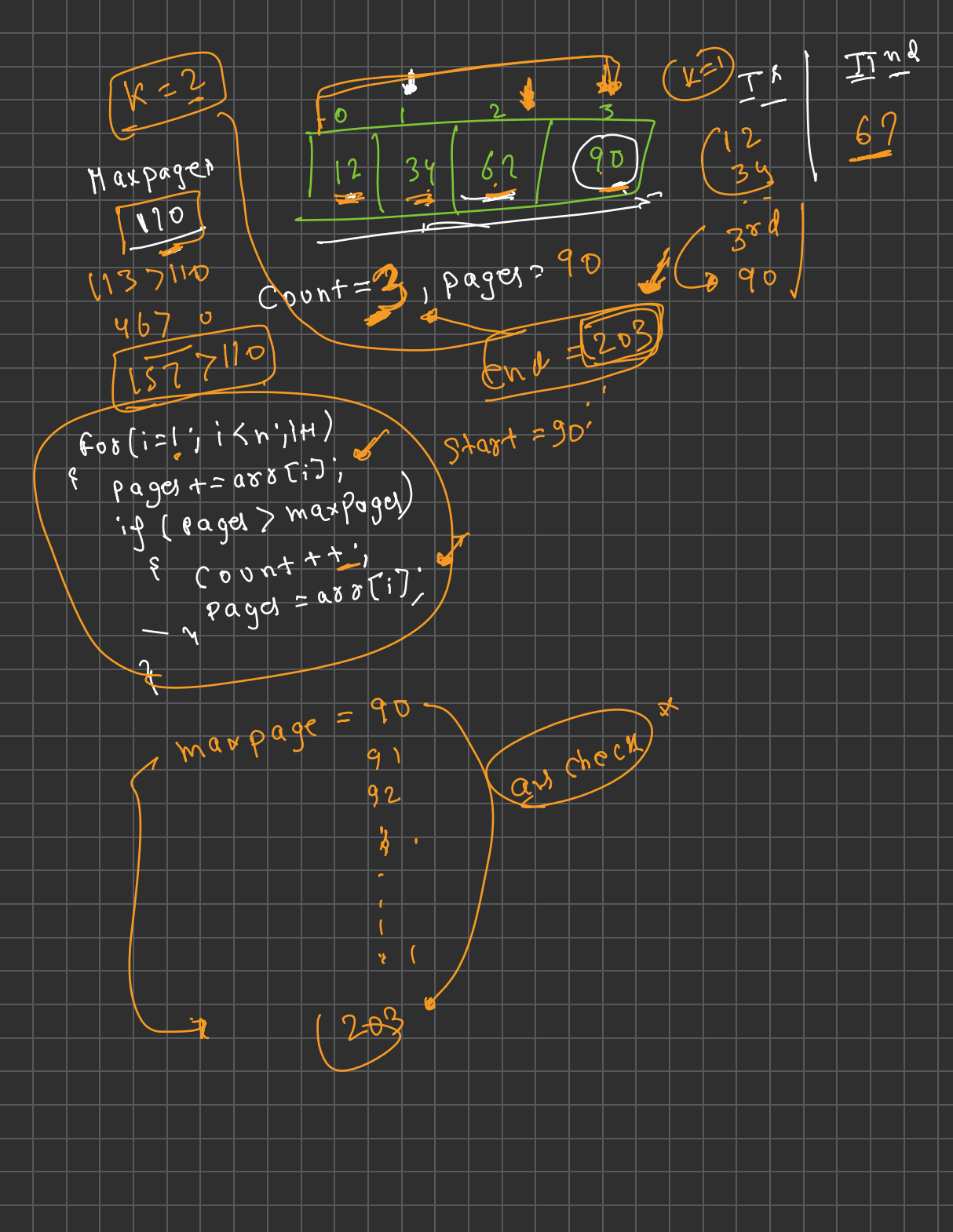
90 90

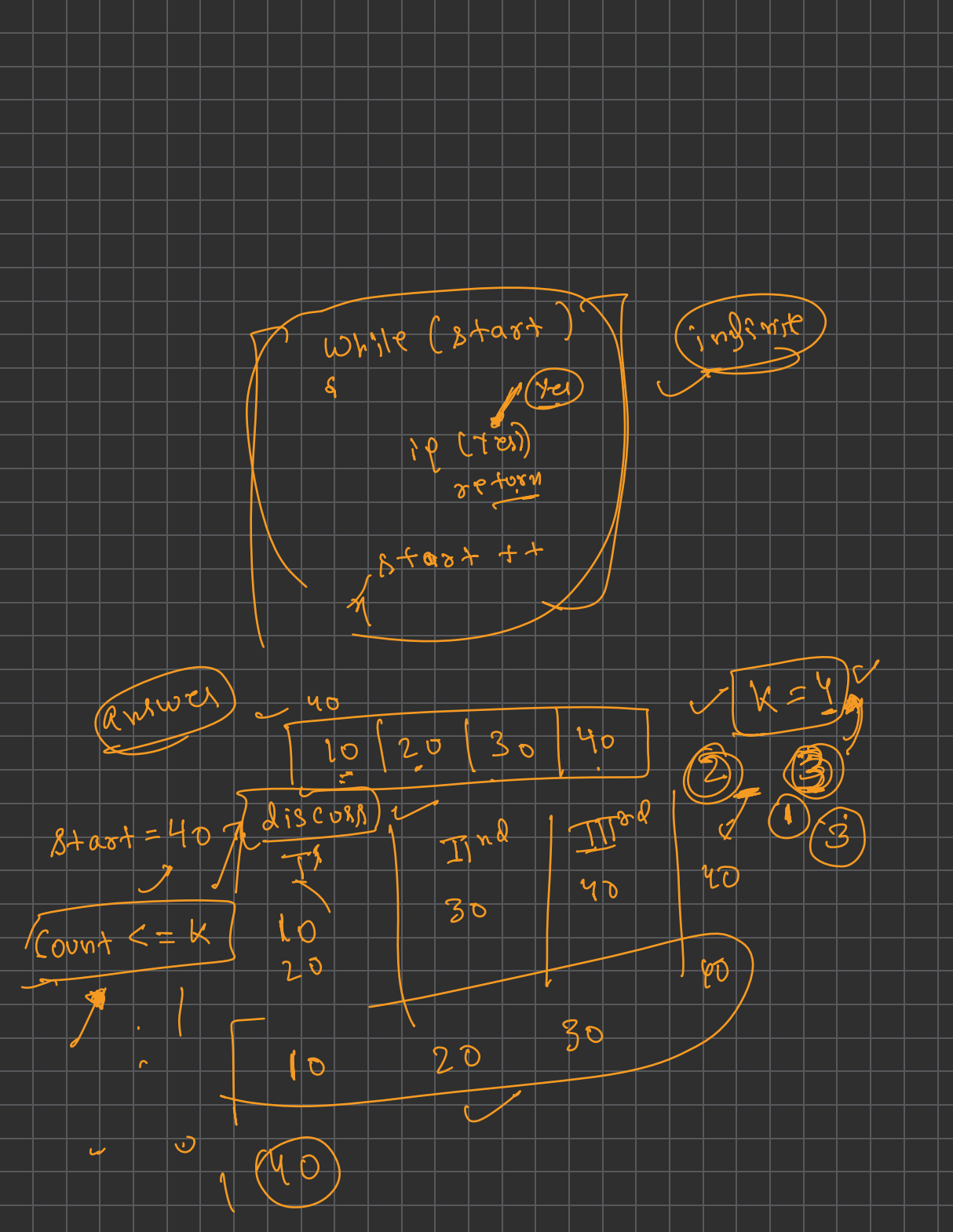
191

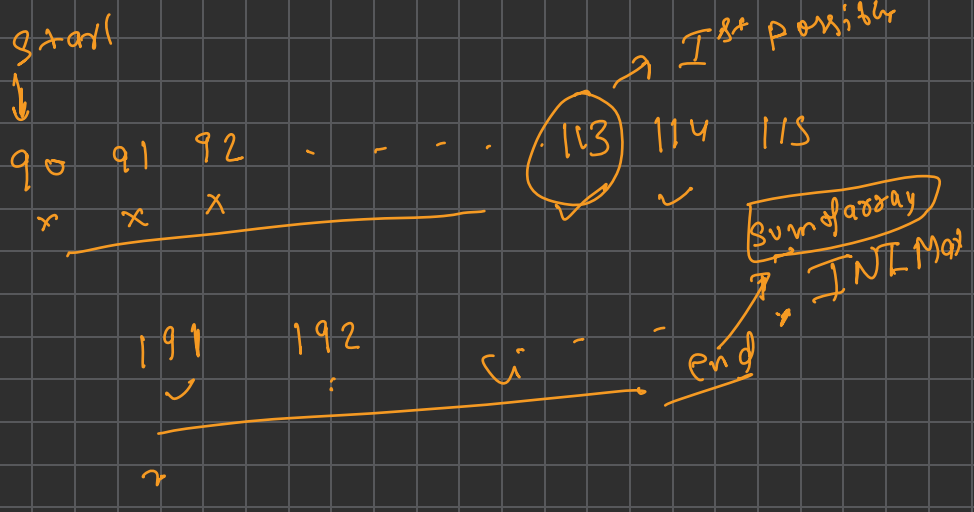
157

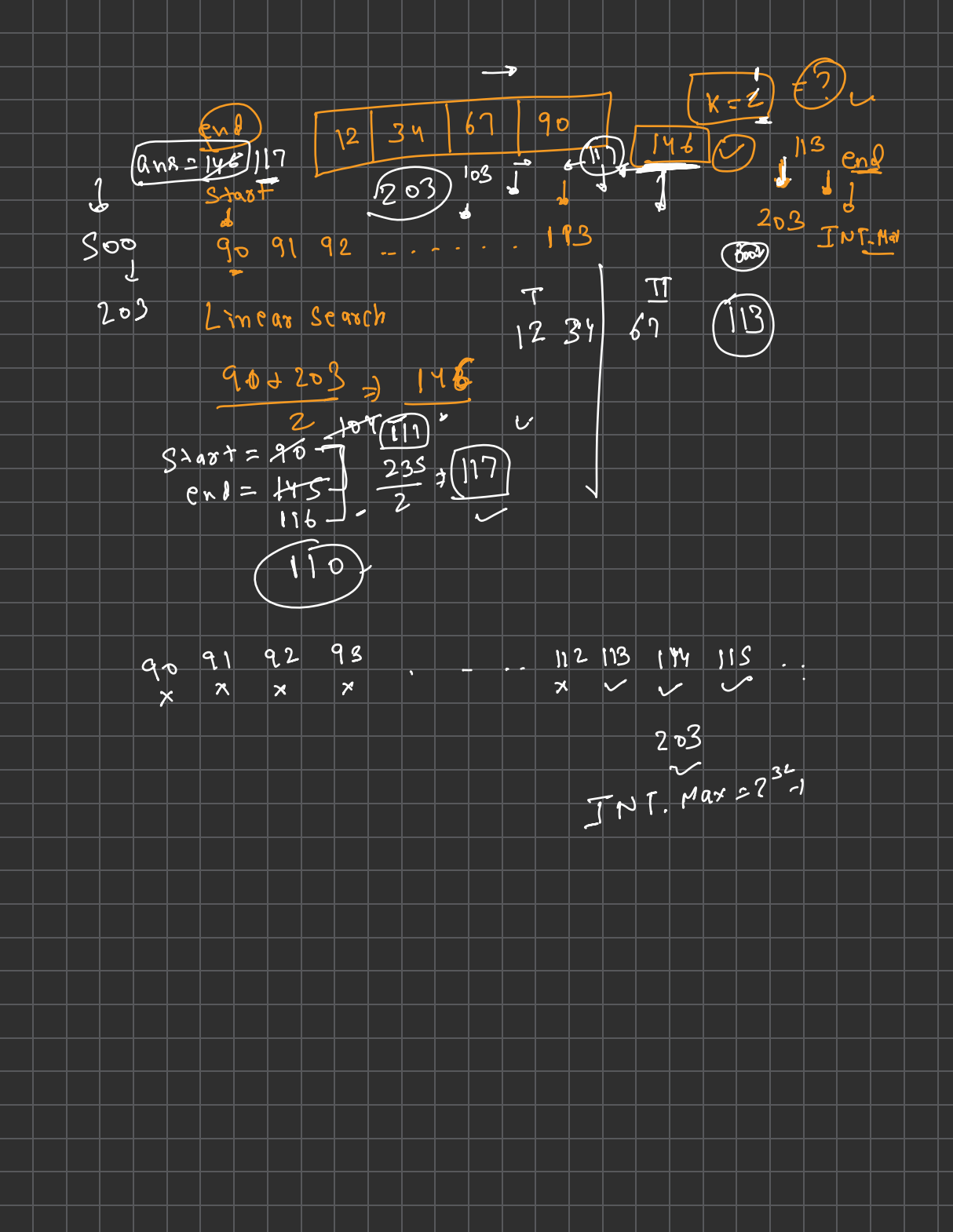
113

113 answer


K = 2
Maxpages
110
113 > 110
467 0
157 > 110
0 1 2 3
12 34 67 90
K=1
I st
12
34
II nd
67
3rd
90
Count = 3 ; pages = 90
end = 203
start = 90
for(i=1; i<n; i++)
{ pages += arr[i];
if (pages > maxPages)
{ count++;
pages = arr[i];
}
}
maxpage = 90
91
92
203
all check


while (start)
{
yes
if (yes)
return
start ++
infinite
answer
40
10
20
30
40
K = 4
discuss
start = 40
count <= K
I st
II nd
III rd
40
30
40
10
20
40
10
20
30
40


start
90 91 92 . . . . 113 114 115
1st positive
Summ array
INT_MAX
191 192
Ci
end


ans = 146
117
start
end
12
34
67
90
203
103
117
146
k = 2
113
end
203
INT_MAX
500
203
90 91 92 193
Linear Search
T
TT
12 34
67
113
90 + 203 = 146
2
start = 90
end = 145
116
111
235
2
117
110
90 91 92 93 ... 112 113 114 115 ...
x x x x x ✓ ✓ ✓
203
INT. Max = $2^{32}-1$

ans = 70 ✓

10	20	30	40

K = 3 (4)

allocation possible

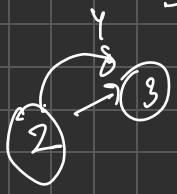
end = mid - 1 ← left

start = 40

end = ~~100~~ ~~64~~ 53

mid = 70

I	II	III
10	40	. . .
20		
30		
10	20 30	40



5	10	30	20	15

K = 3 ↑ 3 paint

1 ~~36~~

2 31

3

4 35 ✓

35

I: 5 10, 30 →

II: 20, 15, . .

III: .

40 min

35 min

left

Start

30 31 32 33 35 80

$\frac{110}{2} = 55$

ans = 55

3	2	2	4	1	4

day = 3

I | II | III

3 | 2 2 | 4 → 6

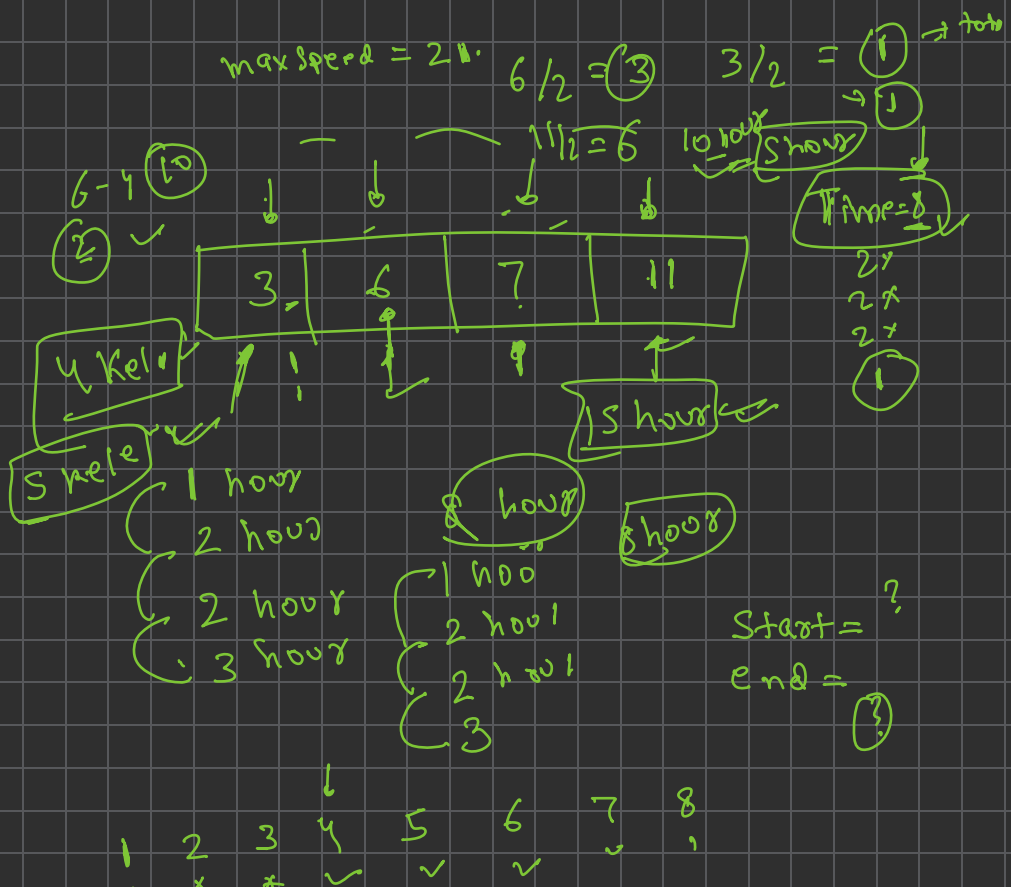
3 | 2 2 4 | 1 4 → 8

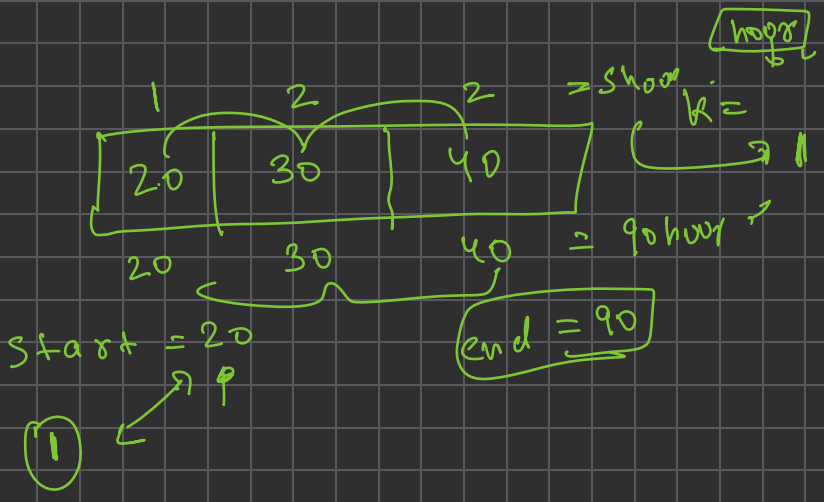
3 2 2 | 4 | 1 4 → 7

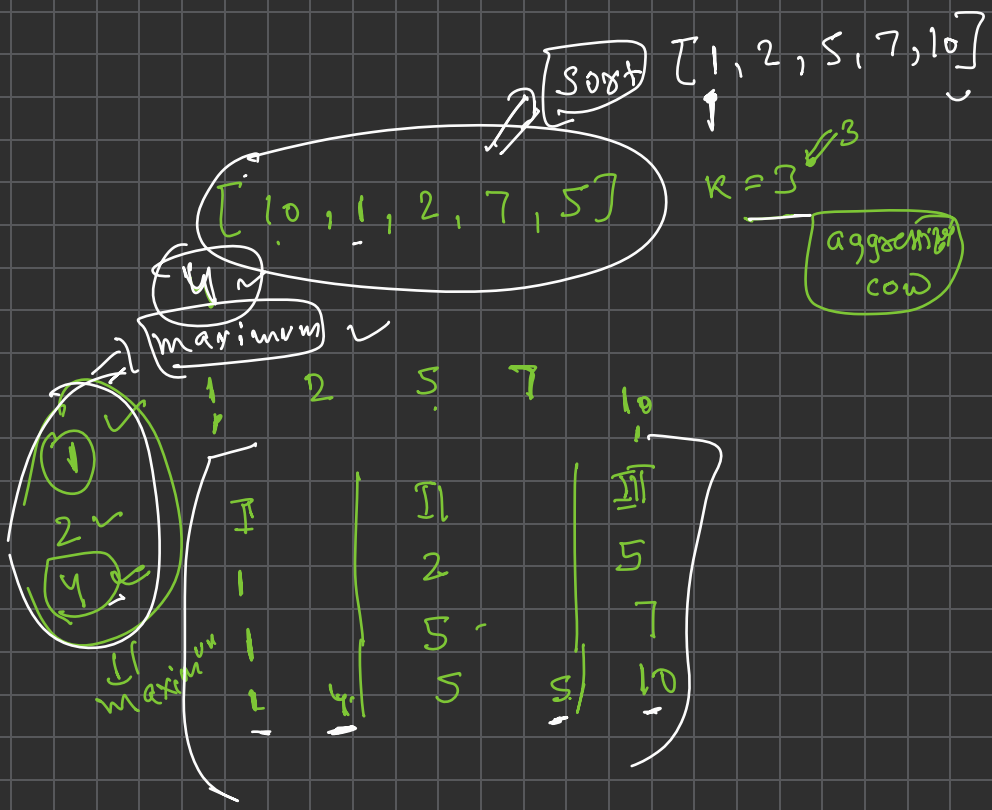
6, 7 pos

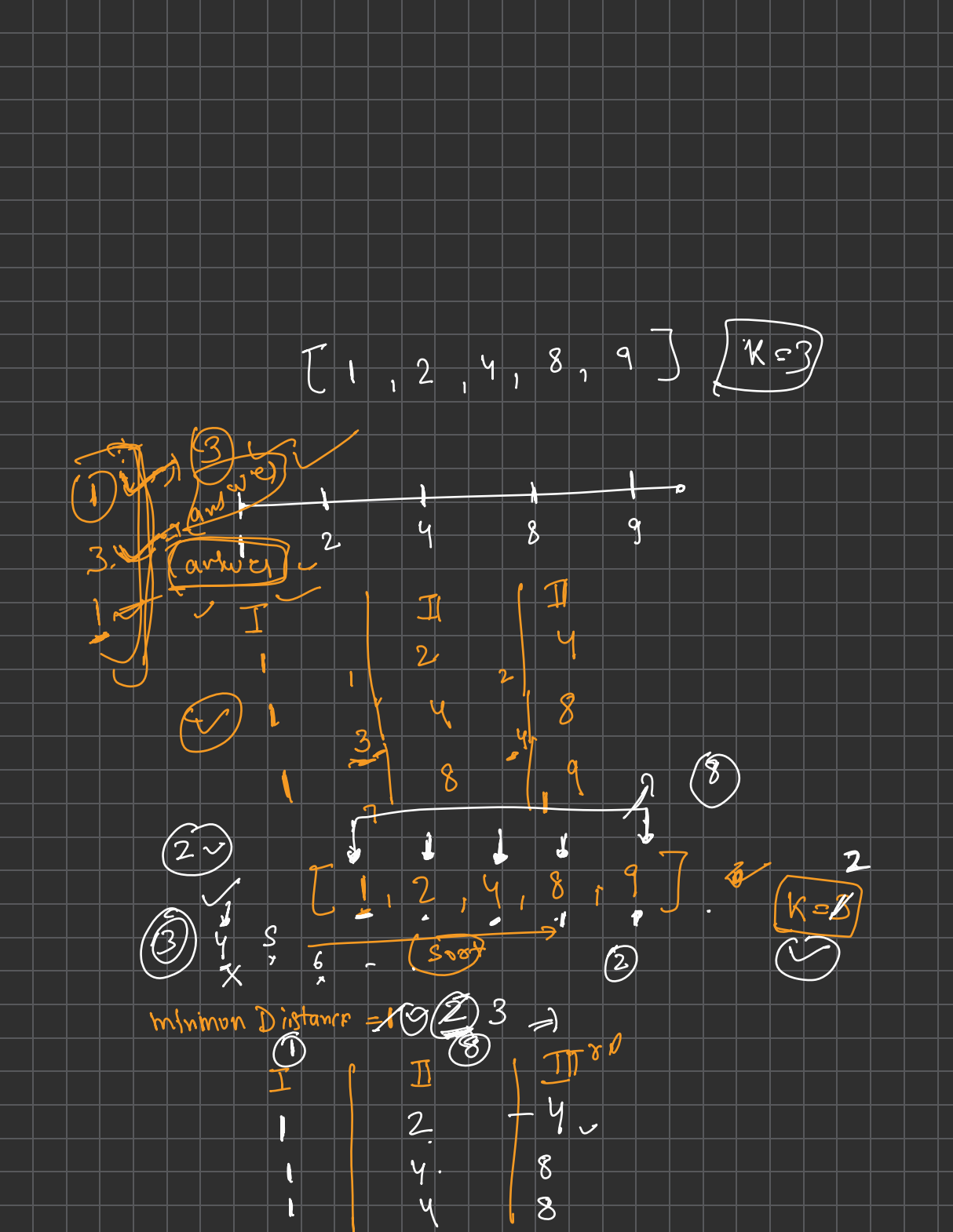
start = 4 → right → ans ← left ← end = 16

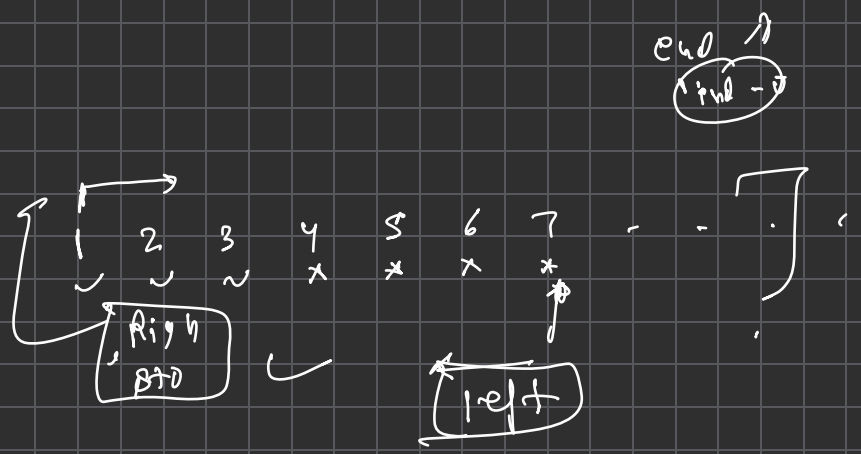
4 5 6 7 8 9 16


maxSpeed = 2
6/2 = 3
3/2 = 1
1 + 1
11/2 = 6
10 hour
5 hour
6 - 4
Time = 0
2 * 2 * 2 * 1
3
6
7
11
4 Kele
5 Kele
5 hour
1 hour
2 hour
2 hour
3 hour
8 hour
8 hour
1 hou
2 hou
2 hou
3
Start = ?
end = ?
1 2 3 4 5 6 7 8
x x x ✓ ✓ ✓


hoop
1
2
2
=show
k=
20
30
40
= 90 hua
20
30
40
start = 20
end = 90
1


Sort
[1, 2, 5, 7, 10]
[10, 1, 2, 7, 5]
k = 3
3
aggressive cow
4
maximum
1
2
5
7
10
1
2
4
I
1
1
1
II
2
5
5
III
5
7
10
If maximum


[1, 2, 4, 8, 9]
k=3
answer
answer
2
4
8
9
I
II
II
4
2
4
8
8
9
[1, 2, 4, 8, 9]
k=3
Sort
minimum Distance =
I
II
III
2.
4.
8
4
8



start = 1

end = ?