I/P = n

O/P = n*(n-1)*(n-2)---1

3

3x2x1

Recursive sol^

① n = n*(n-1)*(n-2)---1 = ⑥

```
fun(n)
{

}
```

② Pick a sub-problem

```
fun(n) {
    sub_sol = fun(n-1)
}
```

③ Combine with orig

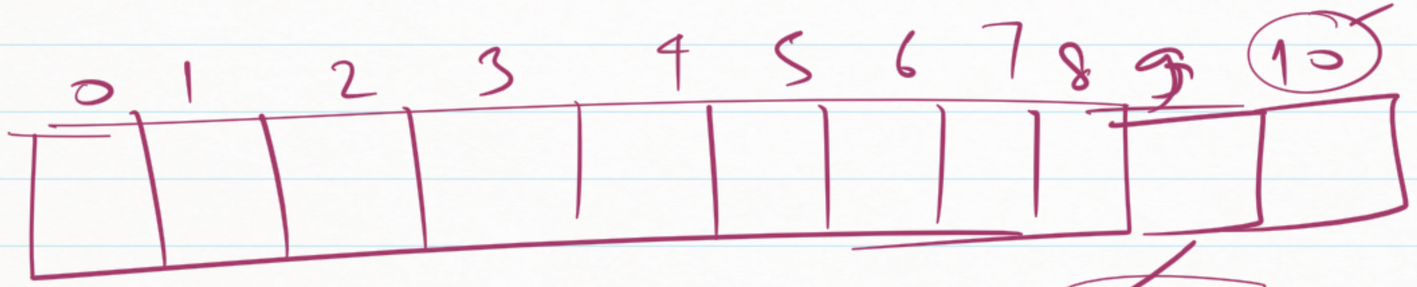
```
fun(n) {
    sub_sol = ''
    return n*sub_sol
}
```

④

```
fun(n)
{
    if n == 1
        return 1
    sub_sol = fun(n-1)
    return n*sub_sol
}
```

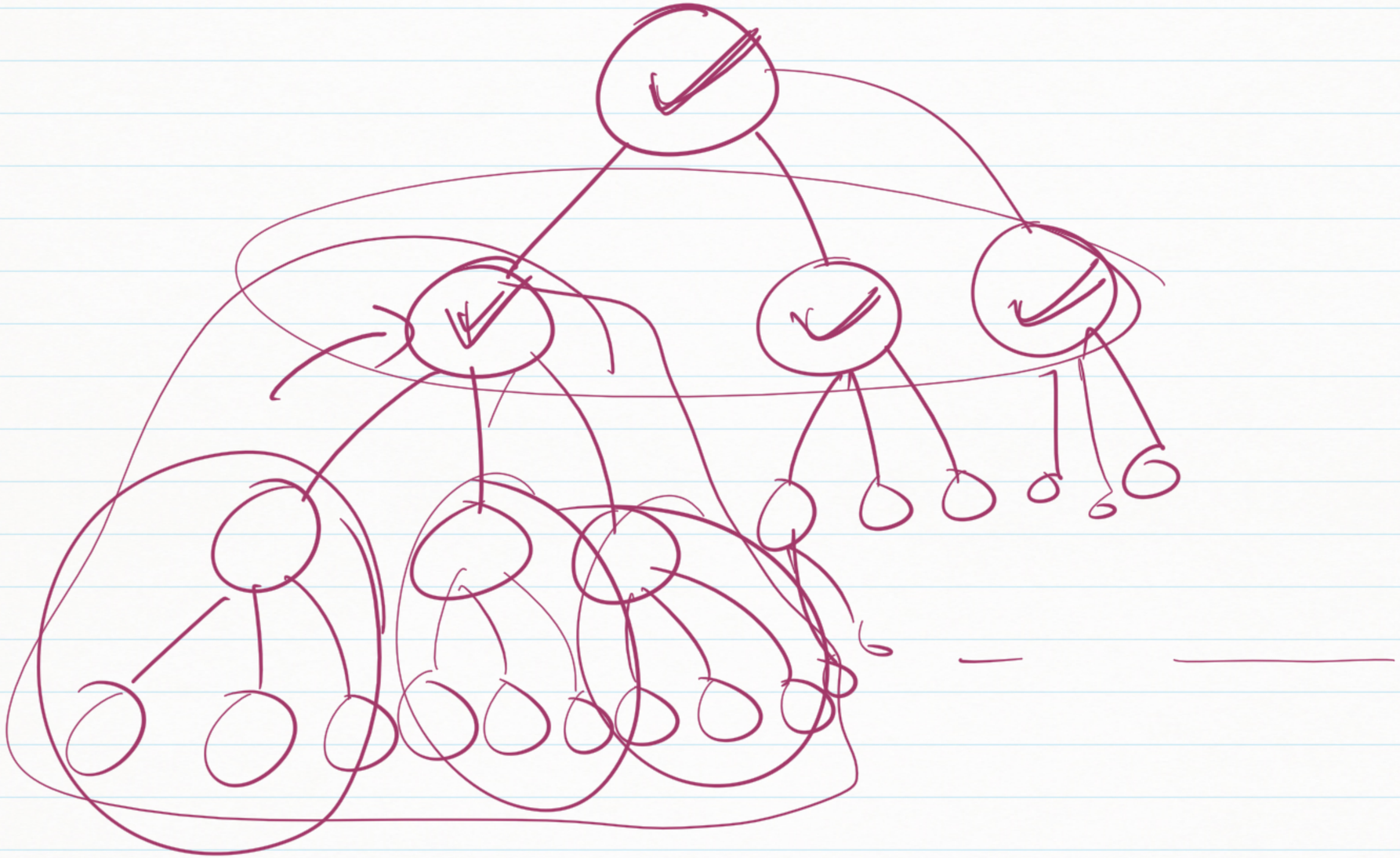
If optimisation involved, we make choices like Max(—,—)

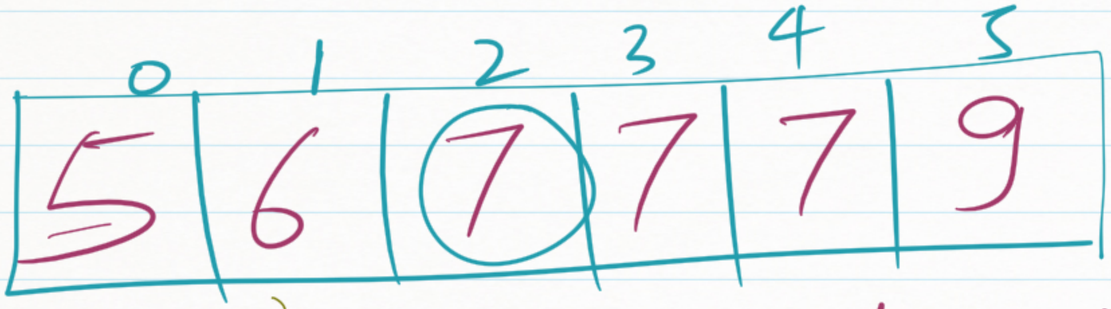
Min(—,—,—)



1, 6, 5

$$\min \begin{cases} \text{MinCoins}(10-1)+1 \\ \text{MinCoins}(10-6)+1 \\ \text{MinCoins}(10-5)+1 \end{cases}$$





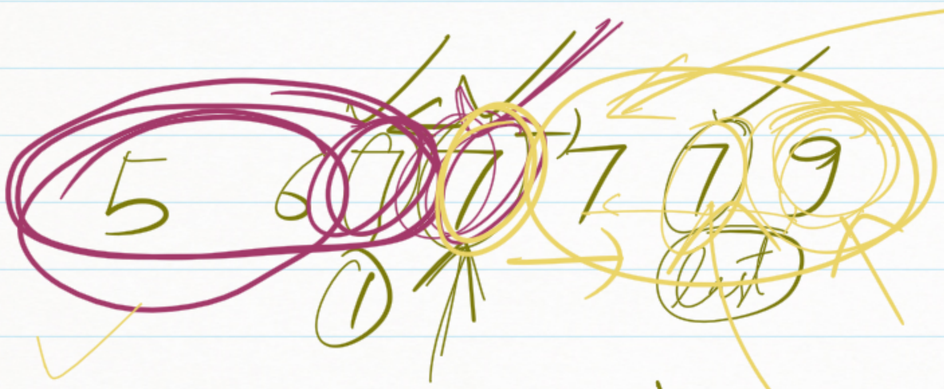
O/P => 1st → 2
last → 4

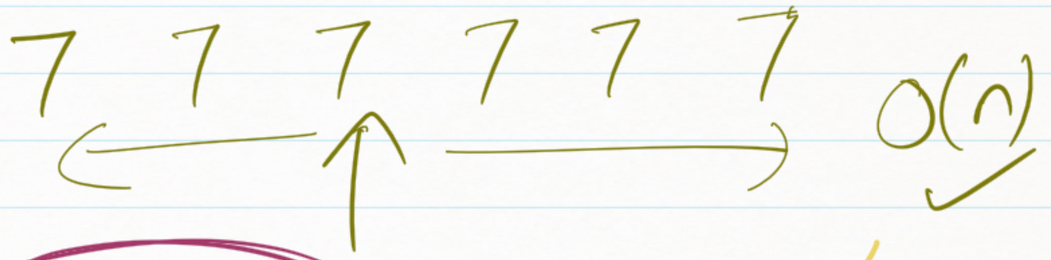
Q1. Find 1st and last occurrence of 7 in an Sorted Array.

5	6	7	7	7	7

O(n) ✓

Binary Search O(log n)






```
binary search (A, n, x)
  { low = 0
    high = n-1,
    while (low <= high)
      { mid = (low + high)/2
        if x == A[mid]
          return mid
        else if x < A[mid]
          high = mid-1
        else
          low = mid+1
  }
  return result }
```

result = -1

1st occur

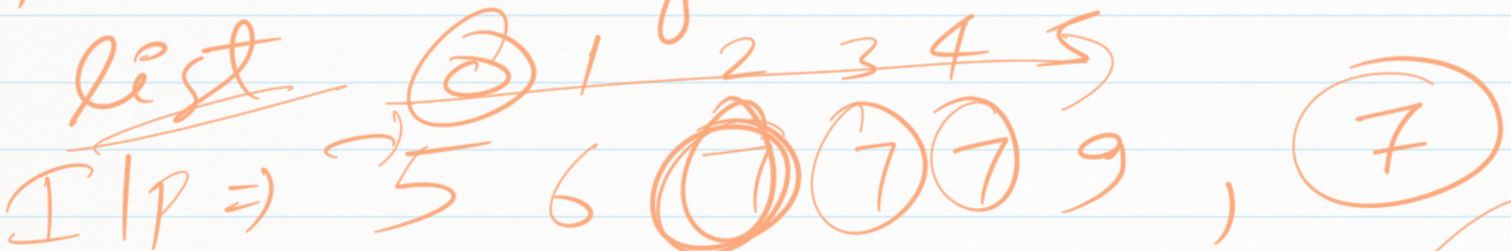
? 2O(log n)

result = mid

high = mid-1

last occur ? low = mid+1

Q Find count of an element in a Sorted list



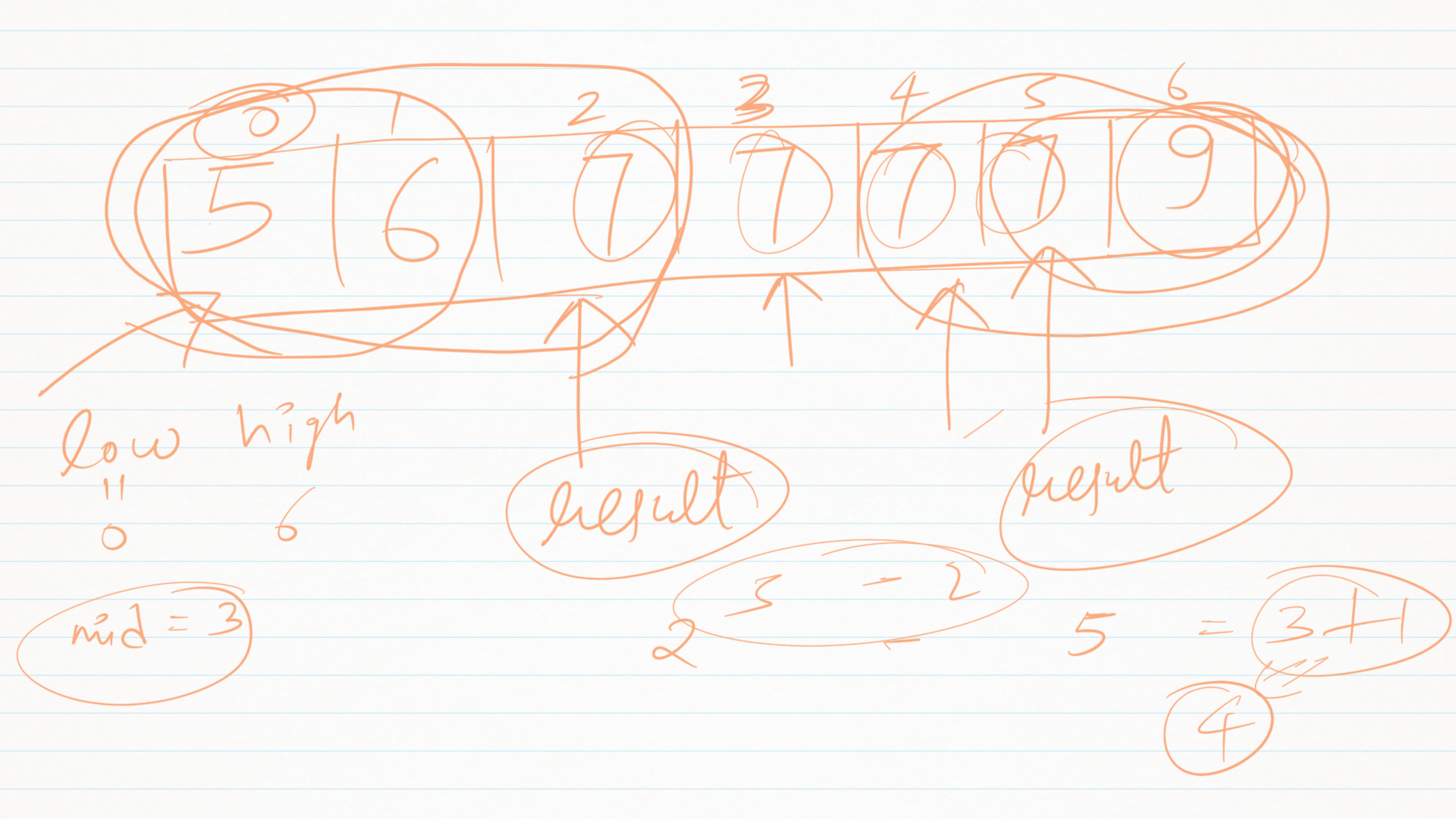
(li - fi) + 1

lastindex = 4

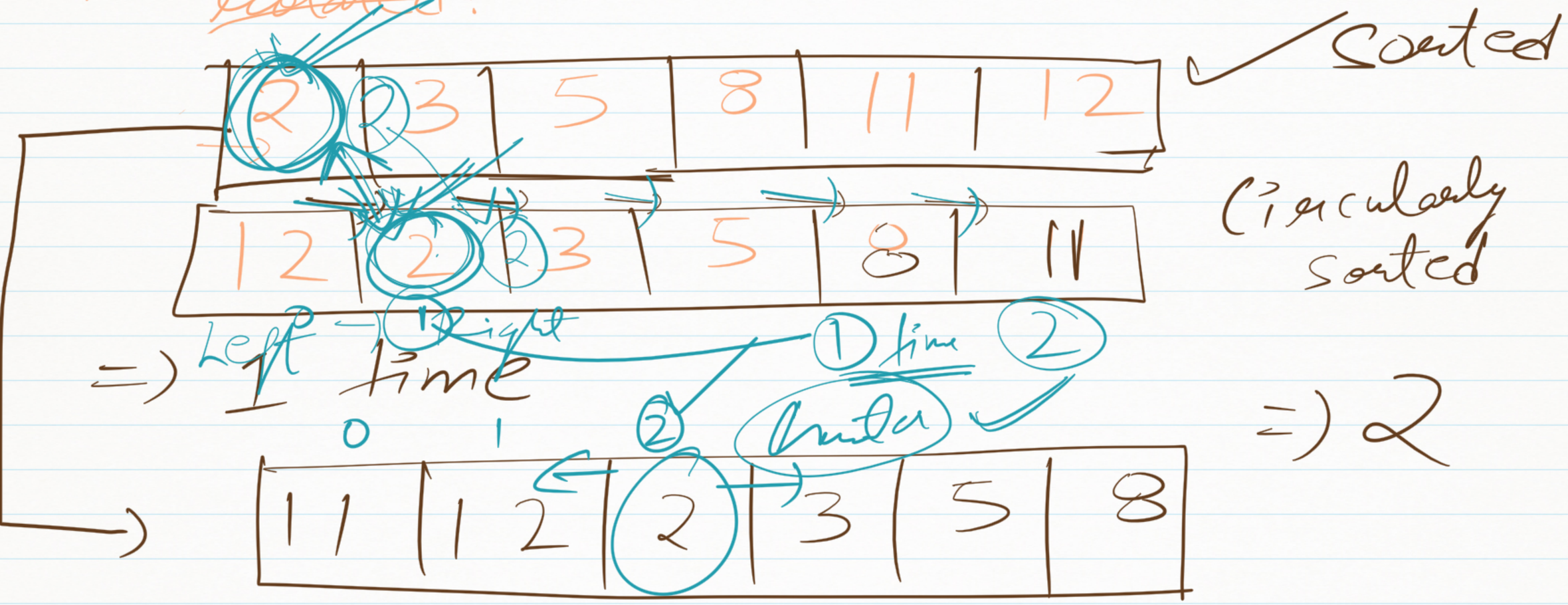
firstindex = 2

(4 - 2) + 1

O(log n)


0 1 2 3 4 5 6
5 6 7 7 7 7 9
low
high
11
0
6
result
result
mid = 3
2
5 - 2
5 = 3 + 1
4

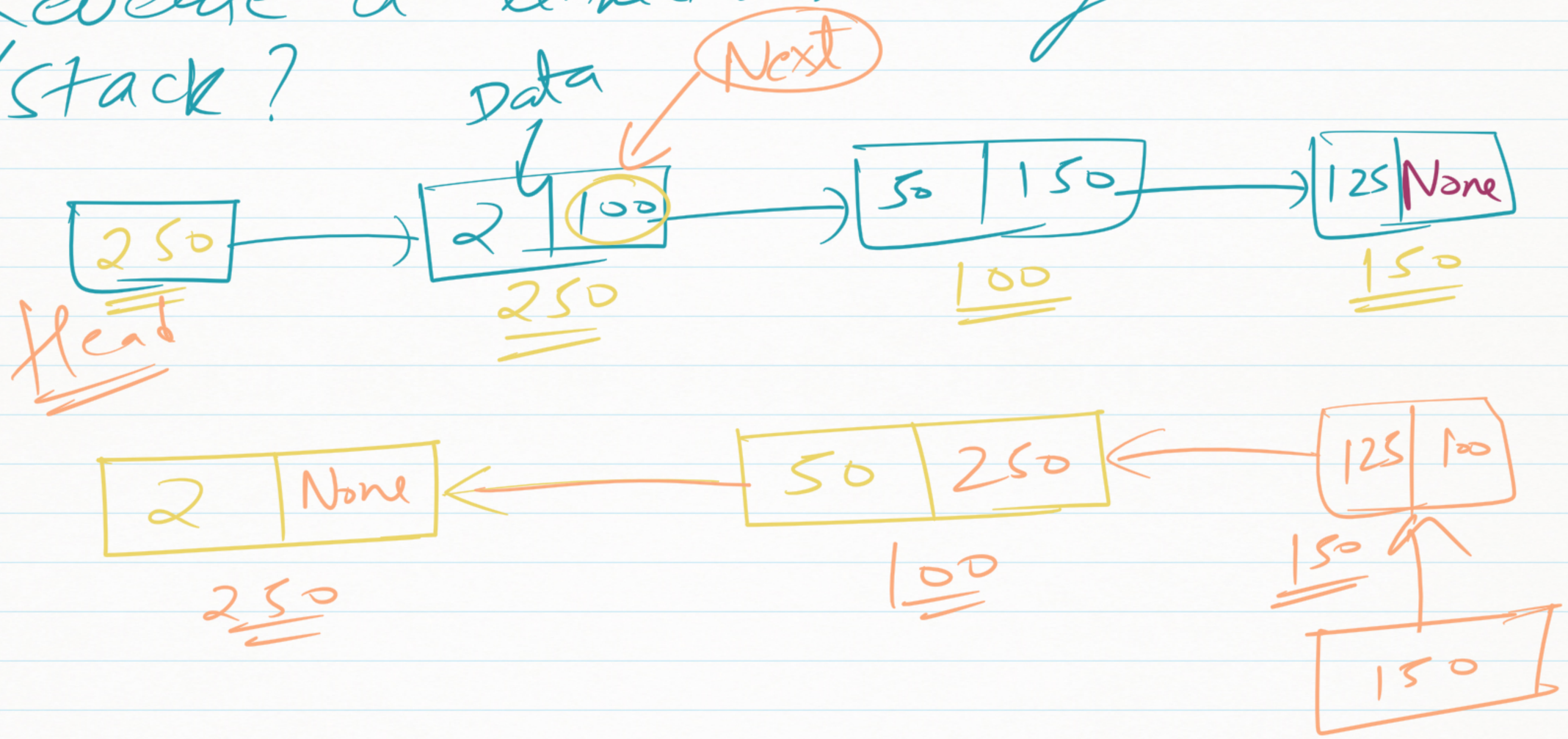
Q How many times is a Sorted Array rotated?

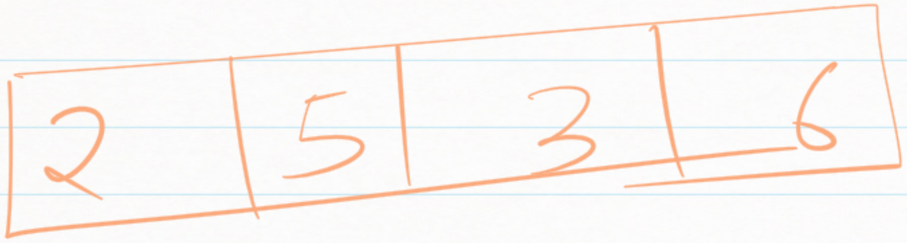


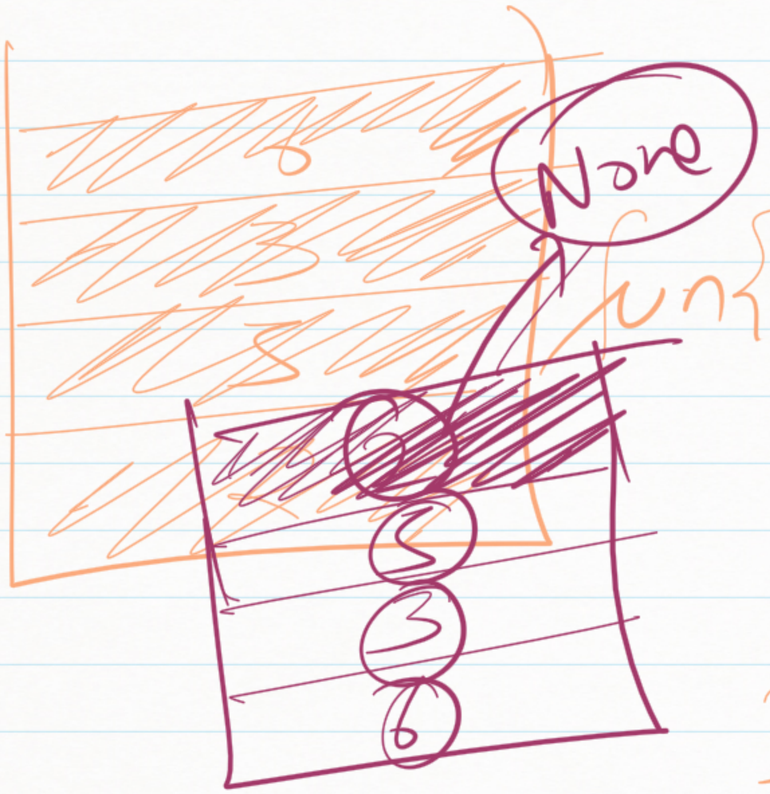
=) Left time

=) 2

Q Reverse a linked list using recursion /stack?







head → 6 → 3 → 5 → 2 → None

```
if (Node.Next != None)
    fun(Node.Next)
print(Node.data)
```

2 5 3 6

Leetcode ✓ Hackerrank - PSA ★

Where do we go from here

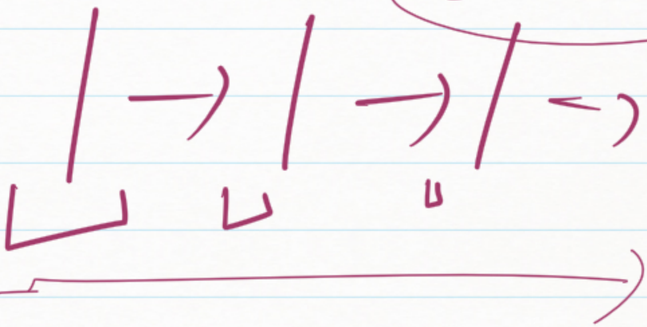
→ Bunch of Algo → ✓

→ Time Comp → ✓

→ ★ Data Struct → ✓

Codeforces ✓

Codechef ✓



★ → Deepen the Understanding of these things ⇒ 2 Question from each