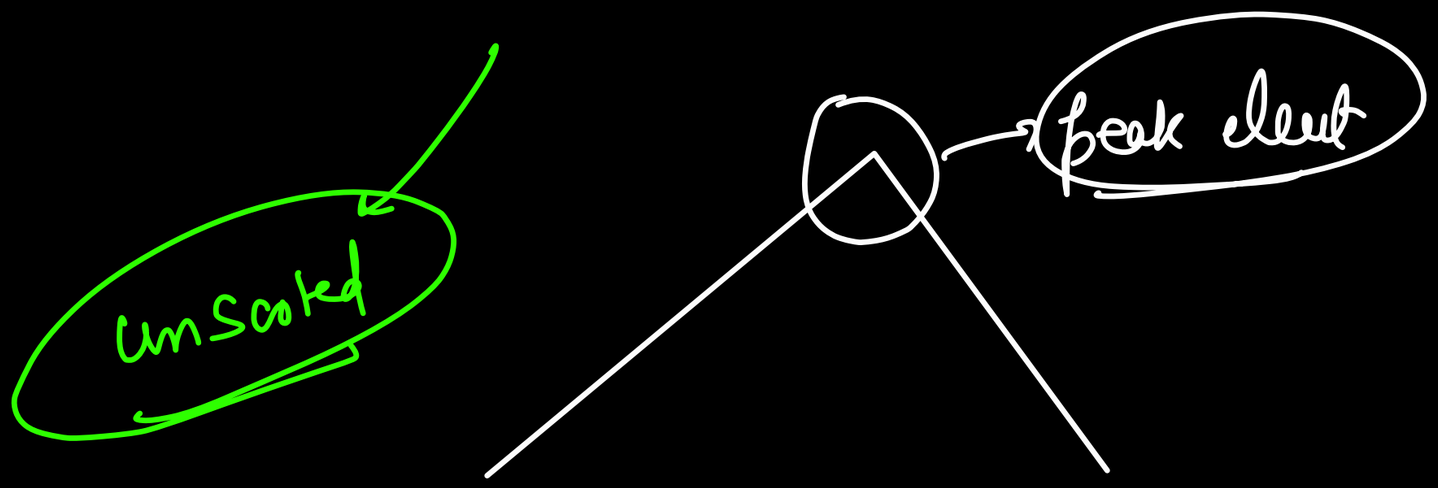
$-\infty$ [1, 2, 1, 3, 5, 6, 4] $-\infty$

1

given array



log n

nums[i-1] < nums[i] > nums[i+1]

↓

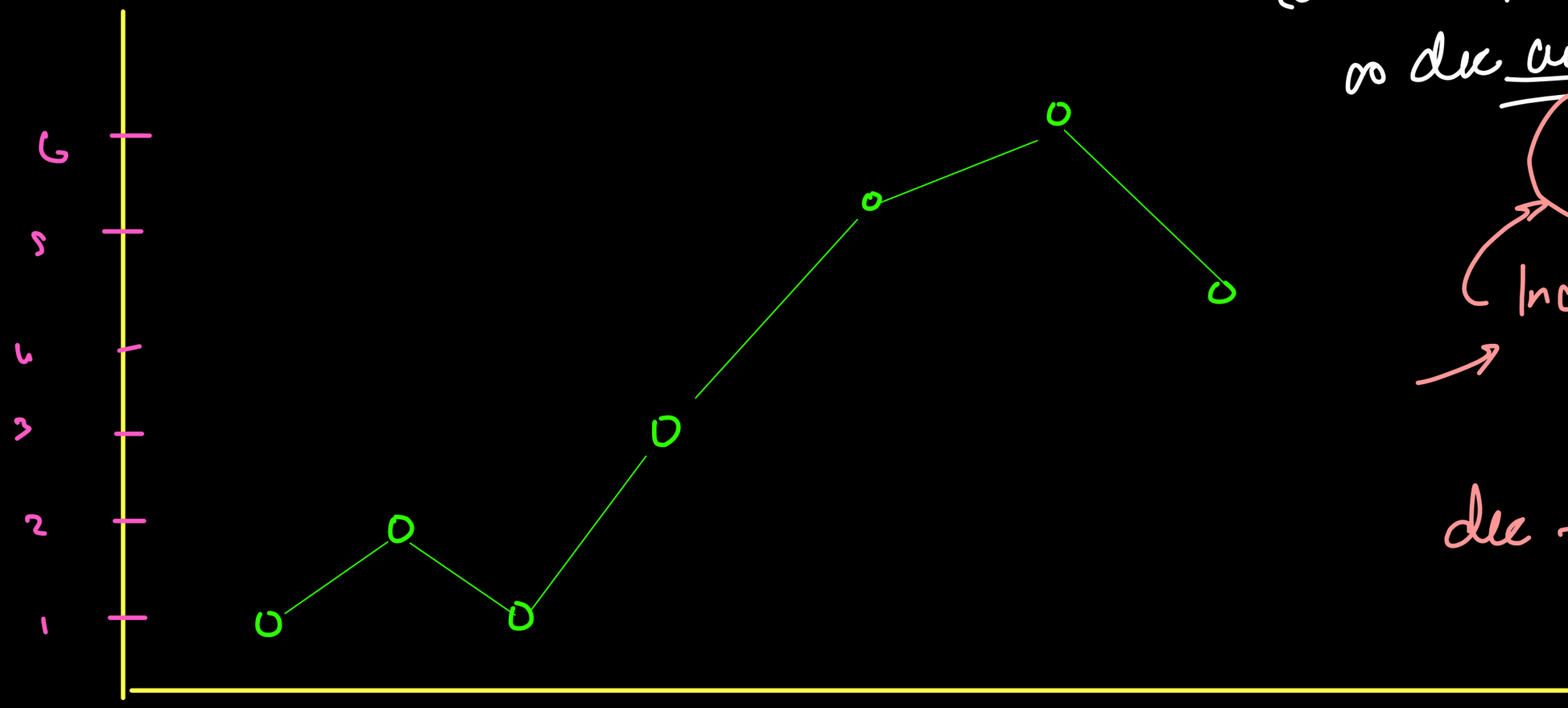
peak

How about we find whether the elem is part of inc or dec seq

(1, 2, 1, 3, 5, 6, 4)

0 1 2 3 4 5 6

mid, lo, hi



any elem can be consider part of any inc or dec seq

if a[mid] < a[mid+1]

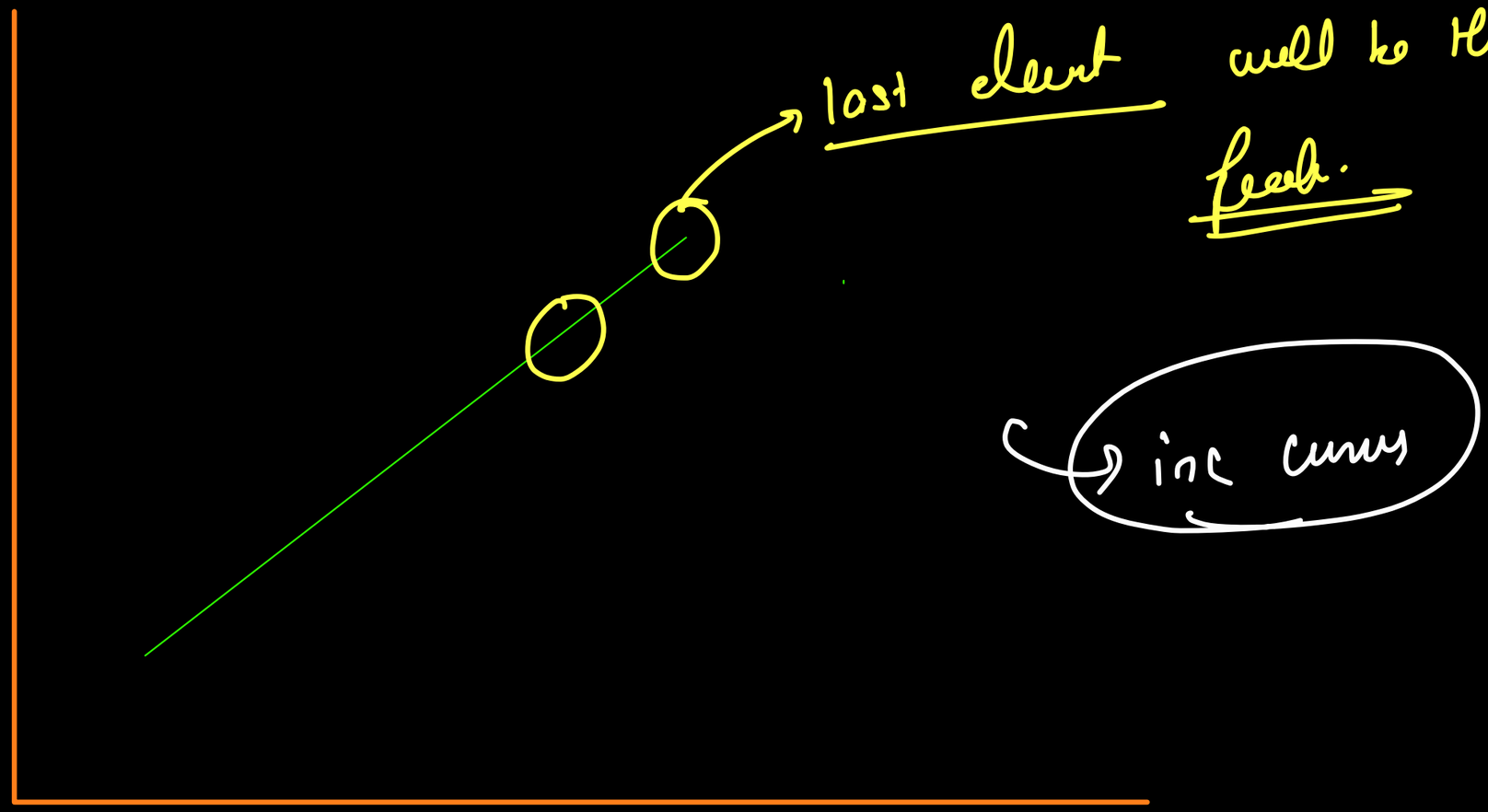
inc → right ✓

left ✗

dec → left ✓

right ✗

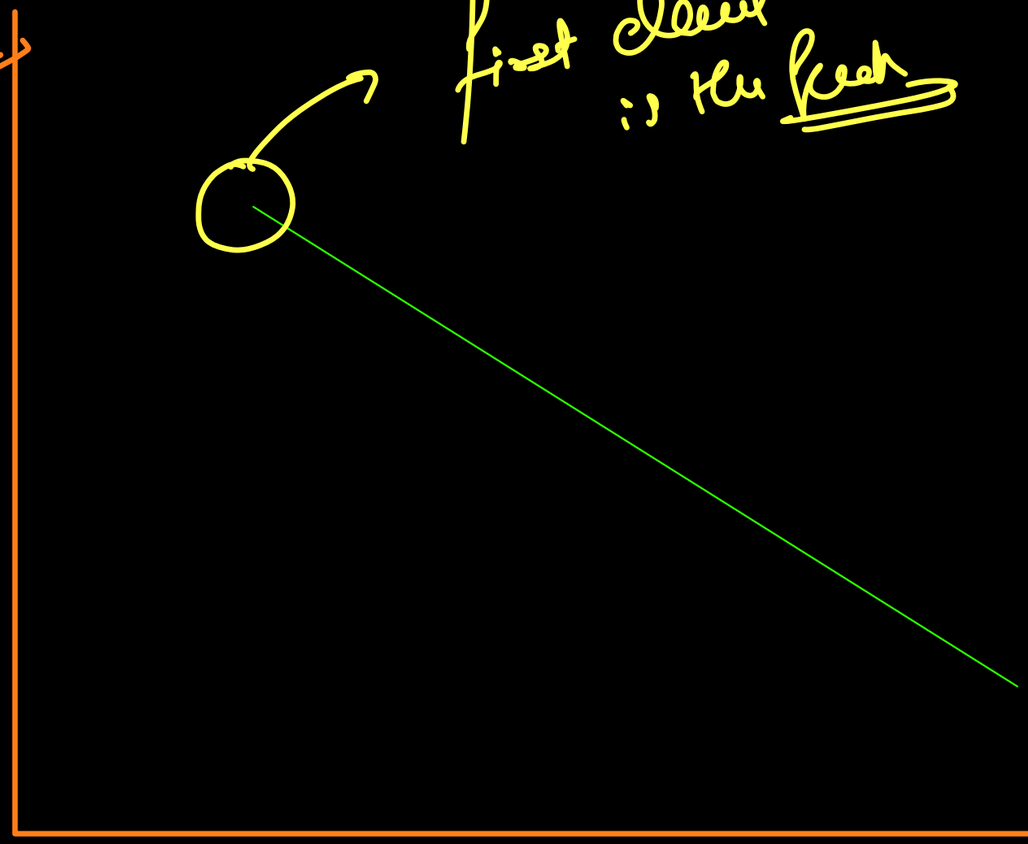
sorted array



1, 2, 3, 7, (5) → ✓ will be the last elem.

↳ inc curve

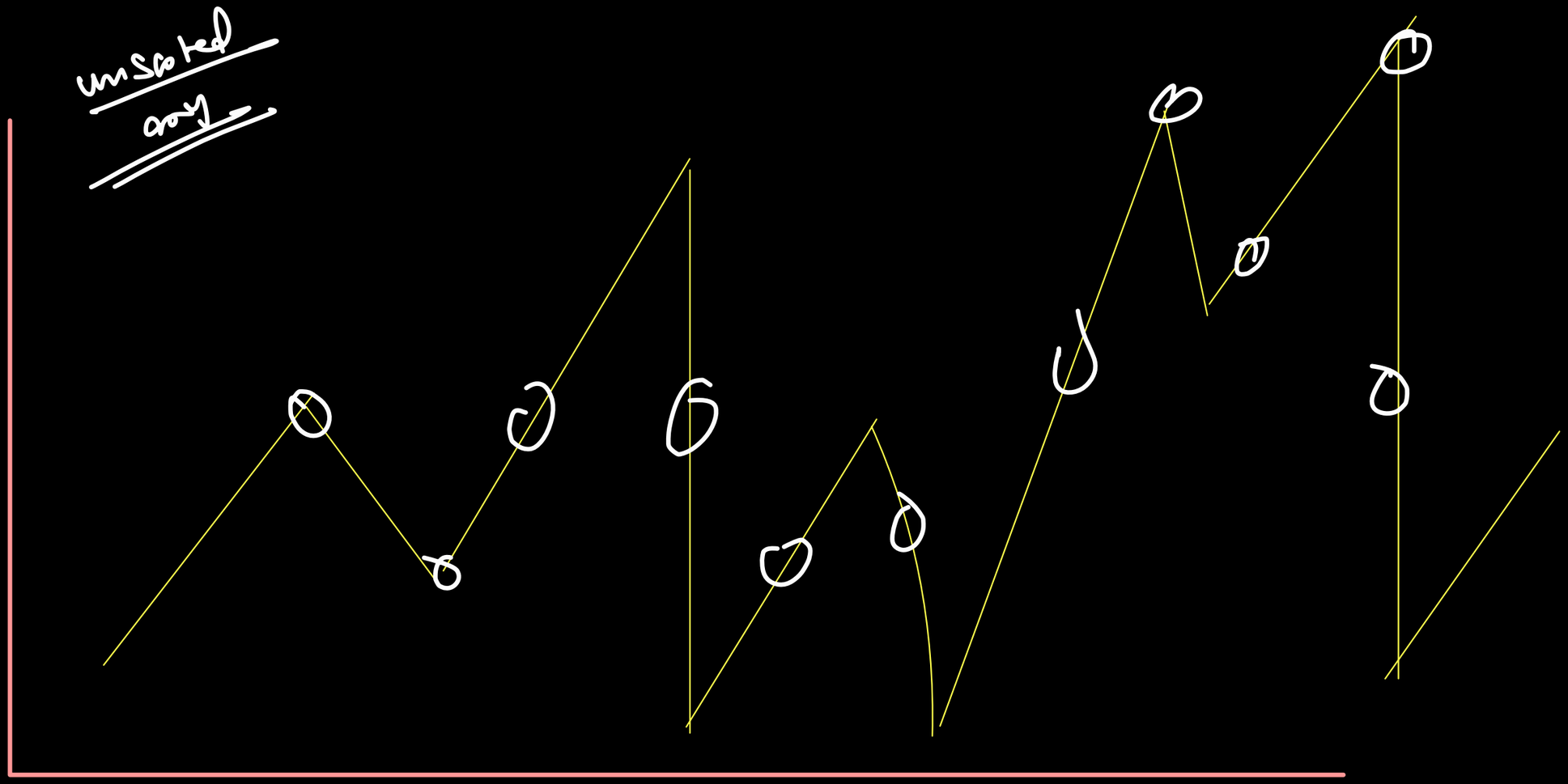
Sorted array



first decent is the peak

(S) 4, 3, 2, 1

→ dec curve



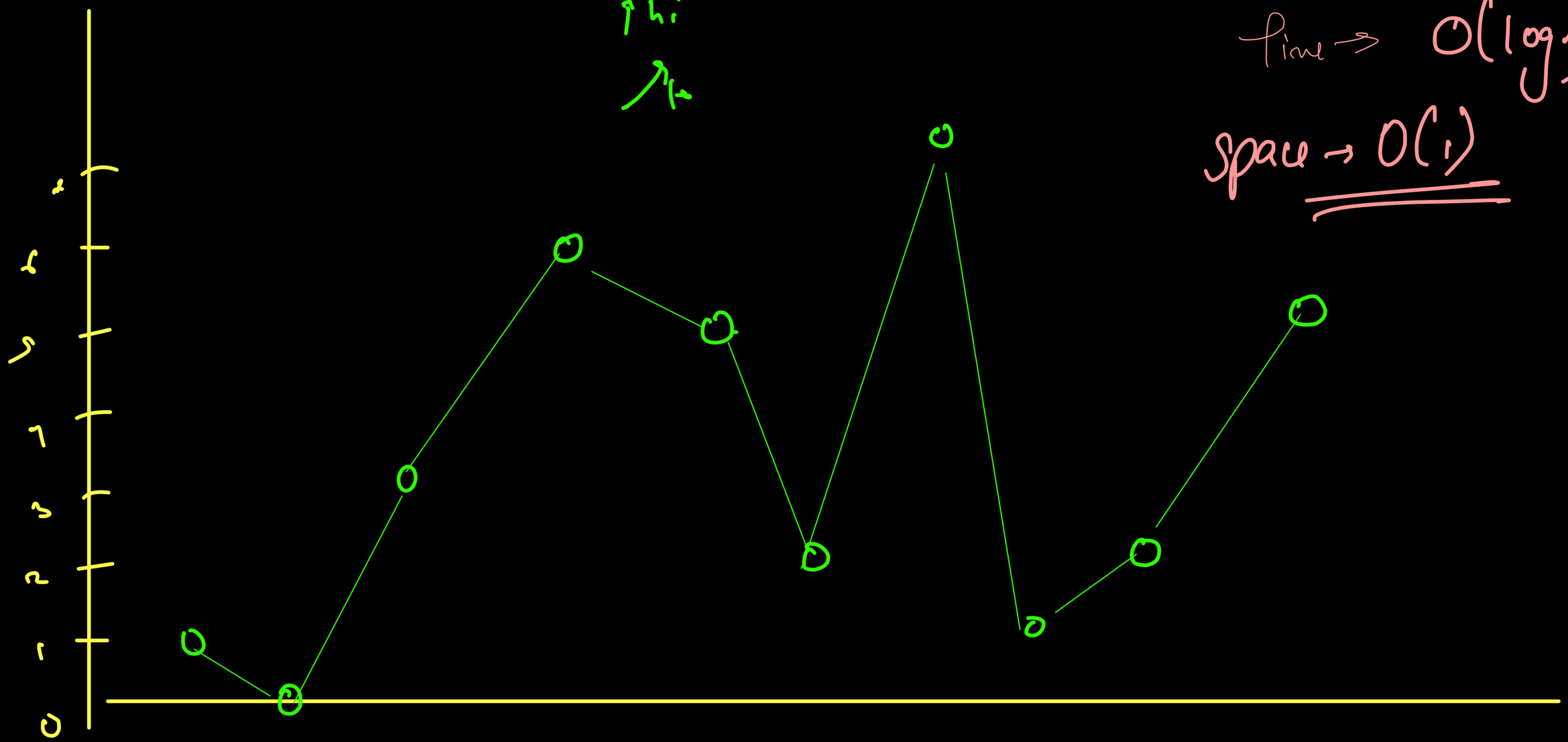
1, 0, 3, 6, 5, 2, 7, 1, 2, 5

0 1 2 3 4 5 6 7 8 9

hi

Time → O(log n)

Space → O(1)



```java
int ans = -1;

int l = 0;
int r = nums.length - 1;

while (l ≤ r) {
    int mid = l + (r - l)/ 2;

    if (mid < nums.length - 1 && nums[mid] < nums[mid + 1]) {
        l = mid + 1;
    } else {
        if (mid == 0 || nums[mid] > nums[mid - 1]) return mid;
        // ans = mid;
        r = mid - 1;
    }
}

return ans;
```

defining search space

mid is not the last index

if true then mid is on inc curve

if mid is peak

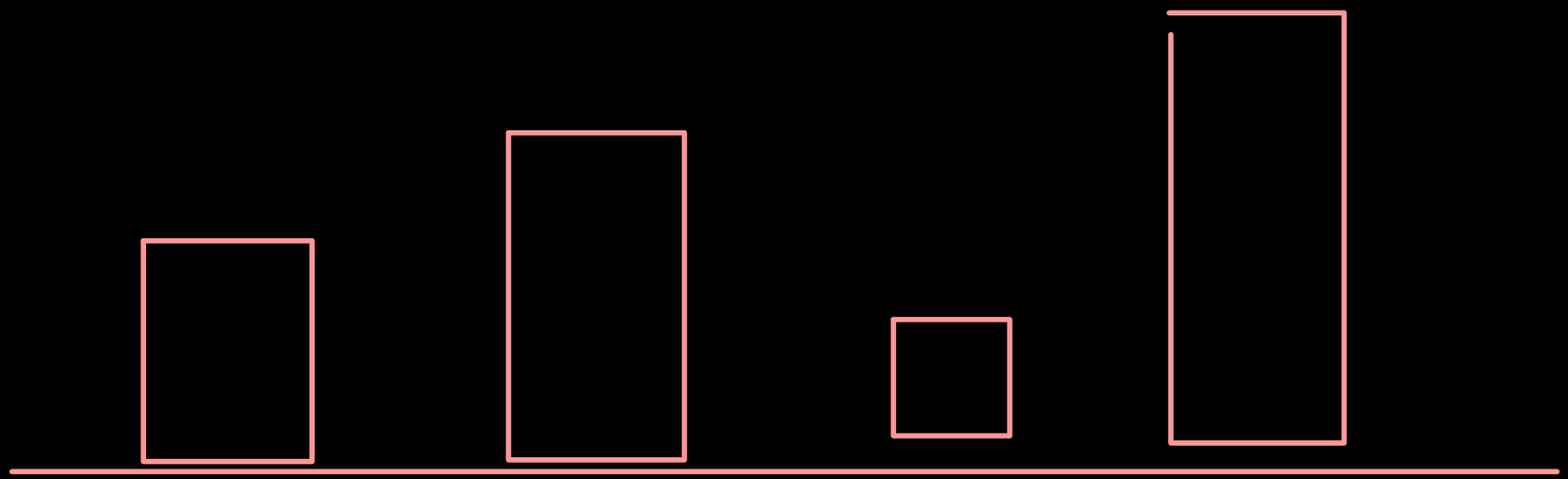
if true we are on dec curve

whether mid is on inc or dec curve?

if ((mid == 0 and nums[mid] > nums[mid+1]) or

(mid == n-1 and nums[mid] > nums[mid-1]) or

(nums[mid] > nums[mid-1] and nums[mid] >
nums[mid+1])



3. 2 → 4
3 → 3

mid → m

ceil (pile[i] / m) → no. of hours req to finish in pile

Koko Bananas

[3, 6, 7, 11] $h = 8$

find min value of k → banana per hour eaty speed

Brute force → to try all possible value of k & check if koko can eat bananas or not.

↳ 1 2 . . . . . - - .

Search space

mid → you want to check if Gj

Koko mid bananas per hour

Speed koko can eat all

bananas or not?

YES

maybe mid is our ans

or we can find better

value < mid

NO

< mid value are useless

go to > mid speed

lo → min speed of koko → 1 banana per hour

hi → max speed → max(piles)

mid

n = 8

3, 6, 7, 11

lo = 1 hi = 11 mid → 6

lo = 1 hi = 5 mid → 3

lo = 4 hi = 5

$\lceil \frac{3}{3} \rceil \to 1$ $\lceil \frac{7}{3} \rceil \to 3$

$\lceil \frac{6}{3} \rceil \to 2$ $\lceil \frac{11}{3} \rceil \to 4$

1 — max of piles

$n \times \log(\text{max of piles})$

$O(n + n \log(\text{max of piles}))$

$\rightarrow O(n \log(\text{max of piles}))$

spells → [5, 1, 3] (n)

[1, 2, 3, 4, 5] (m) ← potions

Success = 7

[4, 0, 3]

Find for each spell how many potions are there such that product of spell & potion >= success

Brute force → To form all possible pairs of spells & potion

↳ O(nm) :-(

```
{ for (i=0; i<n; i++) {
    for (j=0; j<m; j++) {
        if (spells[i] * potions[j] >= sum) {
```

≡

```
        }
    }
}
```

spells [5, 1, 3]

potions [1, 2, 3, 4, 5]

success = 7

lcm (ceil (success / spells[i])) [m - i]

5 * 2 >= 7

2 ⟶ >= 2

x ⟶ spells[i]

x * y >= success

O(nlogm + nlogm)

x * y >= 7

lower bound

JOIN THE DARKSIDE

$$y \geq \frac{\text{Success}}{n}$$

$n = 5$

$$y \geq \frac{7}{5}$$

$$y \geq \lceil 1.\overset{\curvearrowleft}{4} - 7 \rceil$$

$$\underline{\underline{y \geq 2}}$$

$$y \geq 7/3$$

$$y \geq \lceil 2.\overline{3} - 7 \rceil$$

$$\underline{\underline{y \geq 3}}$$

[5, 1, 3] $O(n \log n + m \log m)$ [1, 2, 3, 4, 5]

[1, 3, 5]

i

$+ n + m$

$O(n \log n + m \log m)$

[1, 2, 3, 4, 5]

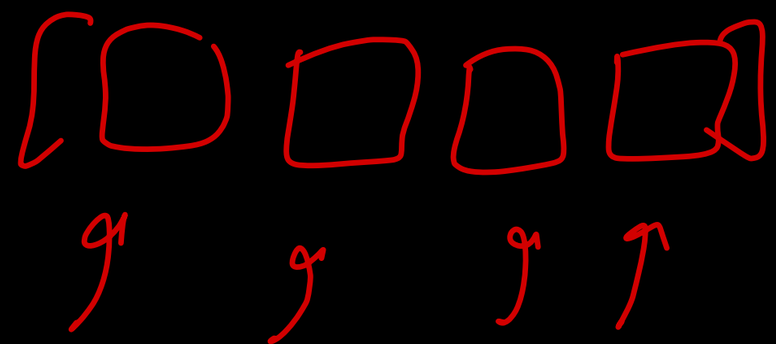
j

Sum = 7

Two pointer

$3 + 5 \geq 7$

$3 \longrightarrow \geq 3$



$\sum s_{1\ 1\ ,3}]$

$[(s,0)\ \ (1,1)\ \ (3,2)$

↳ $[(1,1)\ \ (3,2)\ \ (s,0)]$

within d days

1 day