Binary Search Playlist

Video-25 ←

Facebook
Instagram → codestorywithMIK

Twitter → cswithMIK

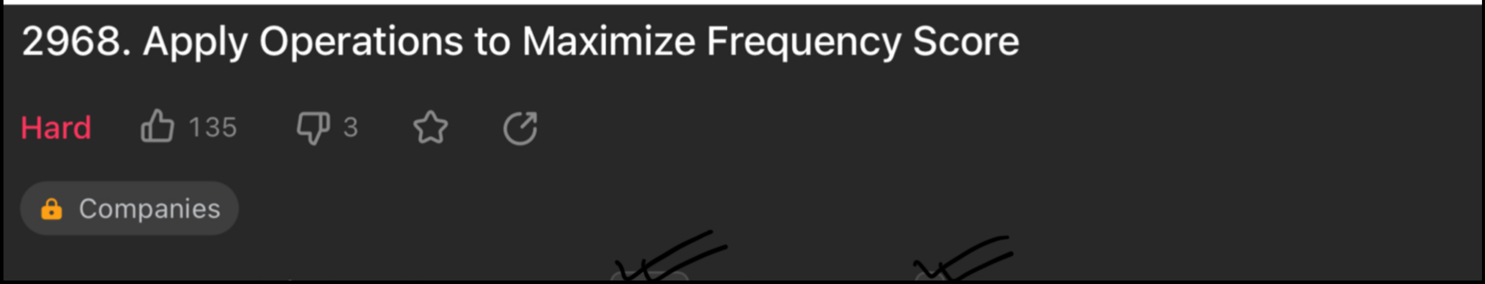
→ codestorywithMIK

weekly contest 376 Q^{n} 4.

Leetcode - 2968

Hard

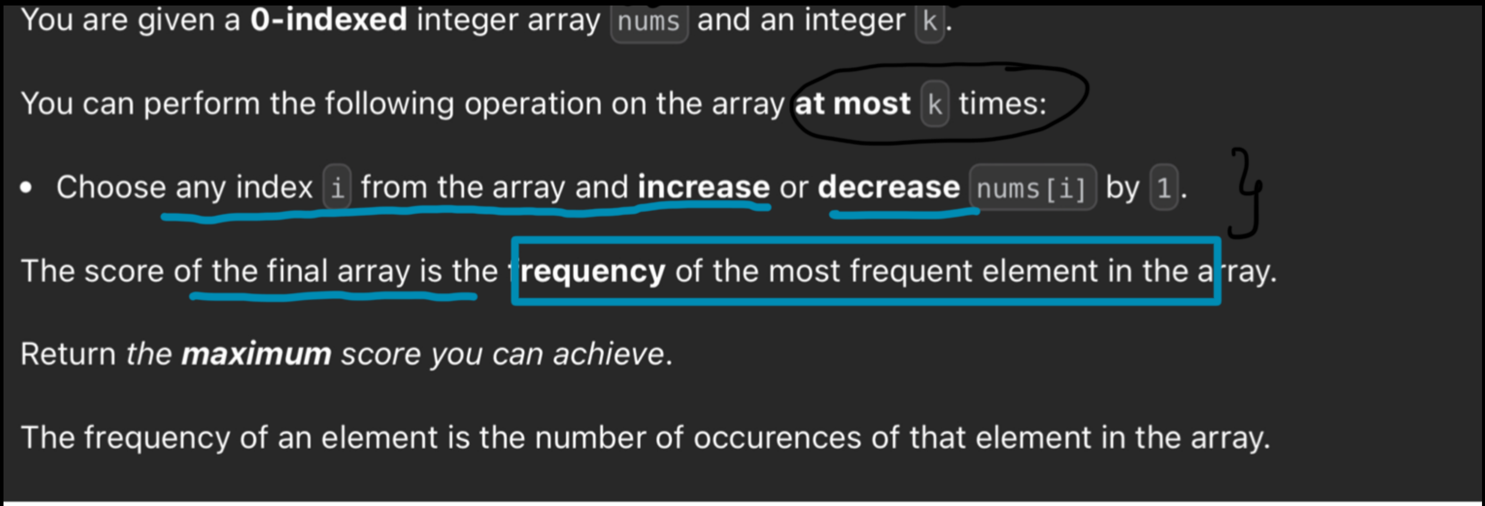
Apply Operations to Maximize Frequency Score



2968. Apply Operations to Maximize Frequency Score

Hard 135 3

Companies



You are given a **0-indexed** integer array `nums` and an integer `k`.

You can perform the following operation on the array **at most** `k` times:

- Choose any index `i` from the array and **increase** or **decrease** `nums[i]` by `1`.

The score of the final array is the **frequency** of the most frequent element in the array.

Return *the **maximum** score you can achieve*.

The frequency of an element is the number of occurences of that element in the array.

Example :- nums = {1, 2, 6, 4} , K = ~~3~~ ~~2~~ ~~1~~ 0

output = 3

{2, 2, 6, 3}

{2, 2, 6, 2}

2 → 3

6 → 1

: Few important Observations

① The target element will be among the one in the array — For minimum operations.

{1, 2, 6, 4} , K=3

oPons 1 $\bigcirc\!\!\!\!0$ 4 2

oPom. 2 1 3

② Sorting

{ 1, 8, 7, 5, 2, 6, 3 }, K=5

Sorting { 1, 2, $\boxed{3, 5, 6, 7, 8}$ }

(•) Sorting

(•) adjacent (closest) elements.
↳ Mini oPos.

Brute Force :-

{1, 2, 6, 4} , K = $\cancel{3}$ 1

Sort { 1, 2, 4, 6 }

1, 2, 4, 6

{1, 2, 6, 6}

score = ~~2~~ 3

$O(N*N)$ → T.L.E.

Most Important Part of this video

$O(N * N)$ → TLE

every element try

calculating opn

Input

“Binary Search on the answer”

Score → max. freq. of an element

min_Score = ~~1~~ ~~3~~ 4 , max_score = 3

l r

mid_score = (min_score + max_score)/2 ; → (1 + 4)/2 = 2

=

K=3

	0	1	2	3
	1	2	4	6
	1	0	2	4

result = 3

which element will be target ??

	0	1	2	3	4	
	1	2	5	6	7	
① ← ops	0	1	4	5	6	→ 16

5 ← opx. 4 3 0 1 2 → 10

⟹ what did we understand???

(•) **Binary Search on answer.**

(•) Sort the array.

(•) min_score = 1 , max_score = n

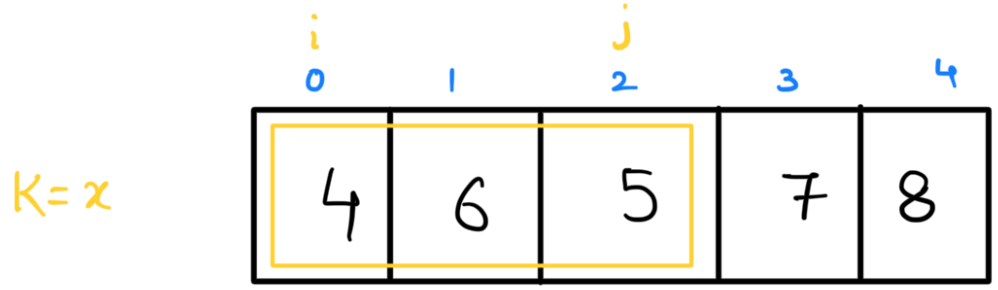
```
while (min_score <= max_score) {

        mid_score = (min_sc + max)/2 ;

        if ( Possible (mid_score, K)) {
            answer = mid_score;
            min_score = mid_score +1;
        } else {

            max_score = mid_score -1;
        }
}
```

Return answer;

Possible (mid-score , K) ;

"We have to check if it's possible to get score = mid-score in nums array"

3



0	1	2	3	4
4	6	5	7	8