






## 3346. Maximum Frequency of an Element After Performing Operations I

Solved 

Medium

 Topics

 Companies

 Hint

You are given an integer array `nums` and two integers `k` and `numOperations`.

You must perform an **operation** `numOperations` times on `nums`, where in each operation you:

- Select an index `i` that was **not** selected in any previous operations.
- Add an integer in the range  $[-k, k]$  to `nums[i]`.

Return the **maximum** possible **frequency** of any element in `nums` after performing the **operations**.

Leetcode  
POTD

Code + PDF  
in Desc.

eg: nums: [1, 4, 5]

k = 1

m = 2

[-1, 1]

↑  
numOperations

nums[i] += [-k, k]    {-1, 0, 1}

[1, 4+1, 5]

[1, 4, 5-1]

5 → ②

4 → ②

①

eg:  $[5, 11, 20, 20]$

$\begin{array}{cccc}
0 & \swarrow & \swarrow & \swarrow & \swarrow \\
& 10 & 6 & 16 & 15 & 25 \\
& & & +0 & +0
\end{array}$

nums[i]

$5+5, 11-1$

$[10, 10] \rightarrow 2$   
 $20, 20$

2

$k=5 \rightarrow [-5, 5]$

$m=1$

$[20, 20]$   
 $\downarrow$   
 2

freq  $\rightarrow$  repetition

20, 20

[20, 5, 10, 20]

$\downarrow$  sort

[5, 10, 20, 20]

$\swarrow$   
Binary  
Search

$\searrow$   
Two  
Pointers

Range

$[-k, k]$



within range

target

$$\begin{matrix} & & & 8 & & & & \pm k \\ & & & 3 & & & & \underline{\underline{\phantom{0}}} \\ 0 & 1 & 2 & & & & & \\ \left[ 5, 10, 20, 20 \right] \end{matrix}$$

$\uparrow$   
 $l$

$nums[l]$

$\checkmark$   
 $20 + 5 \rightarrow 20 \rightarrow T$

$[-k, k]$

$l \rightarrow$  target value

$r \rightarrow$  explore and check which of  
 the elements can converge  
 to my  $nums[l]$

$$① \text{ nums}[l] + k \geq \text{nums}[r]$$

$$② \quad r < n$$



$$\text{nums}[r] \geq \text{nums}[l]$$

while ( $r < n$  and  $l < n$ ) {

while ( $r < n$  and  $\text{nums}[l] + k \geq \text{nums}[r]$ ) {

if ( $\text{nums}[r] == \text{nums}[l]$ ) continue

else { if ( $\text{ops} > 0$ )  $\text{ops}--$   
else break

}

$r++ \rightarrow \text{Expanding}$

}

$\text{ans} = \max(\text{ans}, r - l + 1)$

$l++ \rightarrow \text{Shrinking}$

lets

$\Theta(N)$

Time

}



Sliding window

$k = 6$

5 , 11 20 , 20

l  
→

nums - 1

$$[\underline{88}, 5, \underline{3}]$$

$$k = 17$$

$$m = 1$$

It is not essential that element  
should be a part of array.

$$88 + 17 \quad \times$$
$$- 17 = \textcircled{71}$$

$$53 + 17 = \textcircled{70} \times$$
$$- 17 \times$$

↓  
[ 88, 53 ]

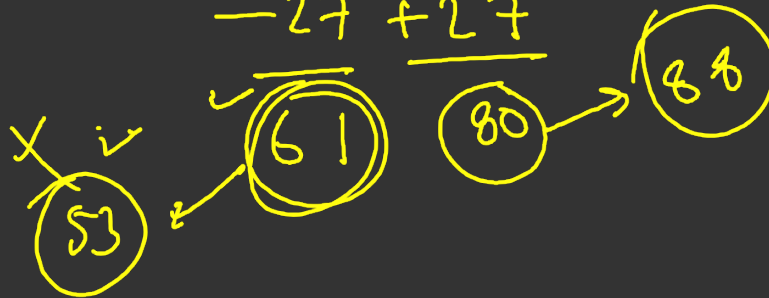
✓  
①

↑  
2

71

27  
—  
88

[ 88, 53 ]  
-27 +27



$$\underline{m=2 \rightarrow 3}$$

$$\underline{k=27}$$

$$[-27, 27]$$

$$\begin{array}{cc} [53, 88] \\ \downarrow \times & \downarrow \times \\ +17 & -18 \\ \hline 70 & 70 \end{array}$$

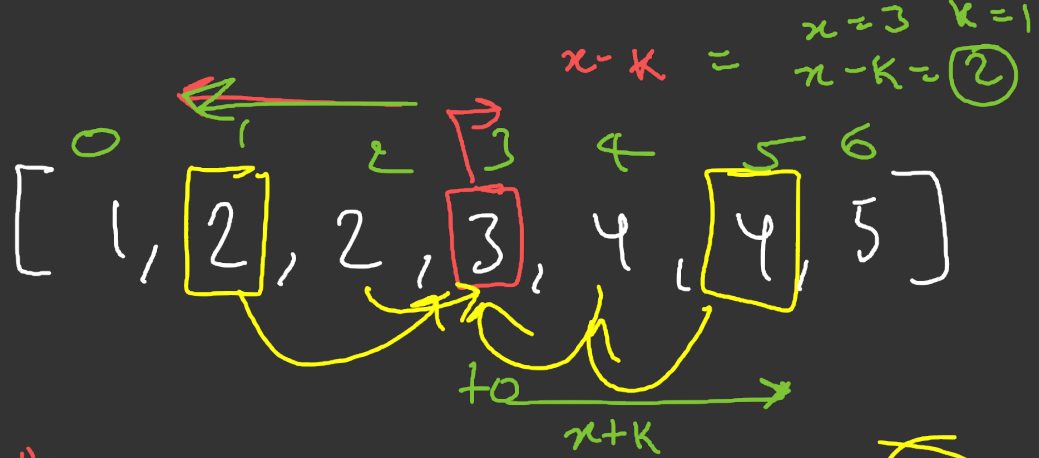
$$[70, 70]$$

70  
target

avg.  
(53, 88)

1, 4, 7, 9, 18, ...

Eg:



$k = 1, m = 4$

$[1, 3, 3, 3, 3, 3, 5]$

5

$5 - 1 + 1 = 5$

$m = 5$  ✓

$m = 4$

$$k=1, m=4$$

$[-1, 4]$

$[1, 2, 2, 4, 4, 5]$

$\uparrow$   $\uparrow$

let's consider all elements  
 $nums[0], \dots, nums[n-1]$

$$x = 3$$

$$x - k$$

2

left

find last value  $\geq x - k$



give me the smallest  
index whose value  
 $\geq x - k$ .

$$x + k$$

4

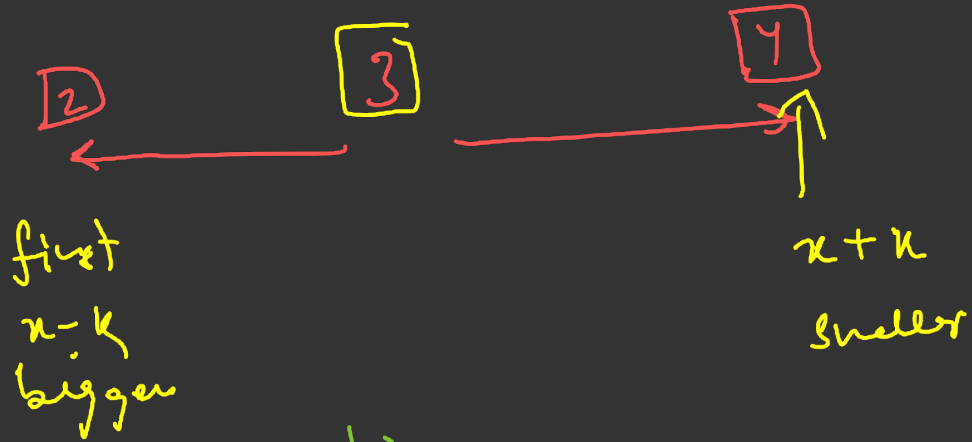
right

find last value  $\leq x + k$

so they have potential

for conversion.

give me largest idx  
whose value  $\leq x + k$



binary search

lower bound

upper bound

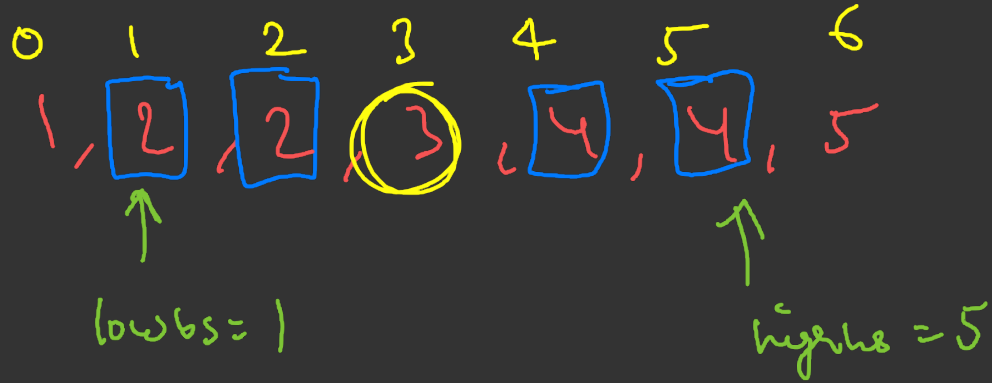


for each  $x$

lowbs = smallest idx with  
value  $\geq x-k$

highbs = biggest idx with  
value  $\leq x+k$

$m \geq 4$



max possible ans =  $highbs - lowbs + 1$

k count.

$$= 5 - 1 + 1 = \textcircled{5}$$

map

3 → 4

1, 2, 2, 3, 3, 3, 3, 4, 5

↑  $x = 3$

↑

- mp[x]

$$\text{freq} = (\text{high bs} - \text{low bs} + 1)$$

freq - subtract

$$\underline{\underline{\text{max possible}}} = (\text{high bs} - \text{low bs} + 1)$$

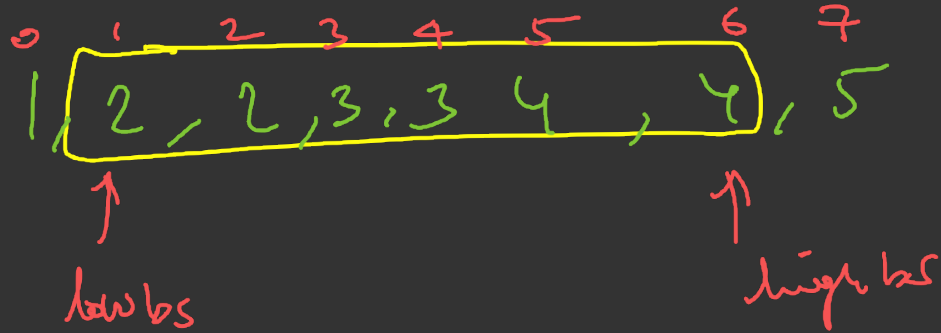


$$\text{mp}[x] \rightarrow 0$$

$$\text{ops} = \min(m, \underline{\underline{\text{max possible}}})$$

$$mp[x] = \underline{\underline{2}}$$

$$m = \cancel{4} \text{ (3)}$$



$$highbs - lowbs + 1 = \text{count} = 6 - 1 + 1 = \text{(6)}$$

$$- mp[x] \quad 6 - 2 = 4$$

$$ops = \min(m, \boxed{\text{count}}) = 3$$

$$\boxed{mp[x] + ops} =$$

$ans = \max ( ans, mp[x] + ops )$

↑ original      ↑ can be converted

return