3397. Maximum Number of Distinct Elements After Operations

Medium | Topics | Companies | Hint

You are given an integer array `nums` and an integer `k`.

You are allowed to perform the following **operation** on each element of the array **at most** *once*:

- Add an integer in the range `[-k, k]` to the element.

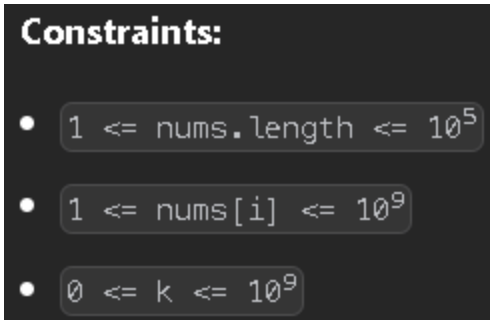
Return the **maximum** possible number of **distinct** elements in `nums` after performing the **operations**.

Example 1:

> **Input:** `nums = [1,2,2,3,3,4], k = 2`
>
> **Output:** 6
>
> **Explanation:**
>
> `nums` changes to `[-1, 0, 1, 2, 3, 4]` after performing operations on the first four elements.

Example 2:

> **Input:** `nums = [4,4,4,4], k = 1`
>
> **Output:** 3
>
> **Explanation:**
>
> By adding -1 to `nums[0]` and 1 to `nums[1]`, `nums` changes to `[3, 5, 4, 4]`.



Python:

```
from typing import List

class Solution:
    def maxDistinctElements(self, nums: List[int], k: int) -> int:
        nums.sort()
        last_picked = -10**18
        distinct_count = 0

        for num in nums:
            lower_bound = num - k
            upper_bound = num + k
            if last_picked < lower_bound:
                last_picked = lower_bound
            else:
                last_picked += 1
            if last_picked <= upper_bound:
                distinct_count += 1
            else:
                last_picked -= 1

        return distinct_count
```

JavaScript:

```
var maxDistinctElements = function(nums, k) {
    if (k === 0)
        return (new Set(nums)).size;

    nums.sort((a, b) => a - b);

    let distinctNums = 0;
    let targetNum = nums[0] - k;
    for (let i = 0; i < nums.length; i++) {
        // If it falls within the range, set nums[i] to it and increment it by 1
        if ((nums[i] - k) <= targetNum && targetNum <= (nums[i] + k)) {
```

```
            nums[i] = targetNum;
            targetNum++;
            distinctNums++;
        }
        // If it falls below the range, first update it to the lowest number in the range
        else if (targetNum <= nums[i] - k) {
            targetNum = nums[i] - k;
            nums[i] = targetNum;
            targetNum++;
            distinctNums++;
        }
    }

    return distinctNums;
};
```

Java:

```
class Solution {
    public int maxDistinctElements(int[] nums, int k) {
        if (k == 0)
            return (int) Arrays.stream(nums).distinct().count();

        Arrays.sort(nums);

        int distinctNums = 0;
        int targetNum = nums[0] - k;
        for (int i = 0; i < nums.length; i++) {
            // If it falls within the range, set nums[i] to it and increment it by 1
            if ((nums[i] - k) <= targetNum && targetNum <= (nums[i] + k)) {
                nums[i] = targetNum;
                targetNum++;
                distinctNums++;
            }
            // If it falls below the range, first update it to the lowest number in the range
            else if (targetNum <= nums[i] - k) {
                targetNum = nums[i] - k;
                nums[i] = targetNum;
                targetNum++;
                distinctNums++;
            }
        }

        return distinctNums;
    }}
```