2616. Minimize the Maximum Difference of Pairs

Attempted

Medium | Topics | Companies | Hint

You are given a **0-indexed** integer array `nums` and an integer `p`. Find `p` pairs of indices of `nums` such that the **maximum** difference amongst all the pairs is **minimized**. Also, ensure no index appears more than once amongst the `p` pairs.

Note that for a pair of elements at the index `i` and `j`, the difference of this pair is `|nums[i] - nums[j]|`, where `|x|` represents the **absolute value** of `x`.

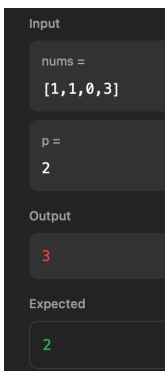
Return *the **minimum maximum** difference among all `p` pairs.* We define the maximum of an empty set to be zero.

Example 1:

> **Input:** nums = [10,1,2,7,1,3], p = 2
> **Output:** 1
> **Explanation:** The first pair is formed from the indices 1 and 4, and the second pair is formed from the indices 2 and 5.
> The maximum difference is max(|nums[1] - nums[4]|, |nums[2] - nums[5]|) = max(0, 1) = 1.
> Therefore, we return 1.

```python
class Solution:
    def minimizeMax(self, nums: List[int], p: int) -> int:
        count = Counter(nums)
        arr = []


        for c in count:
            if count[c] % 2 == 1:
                arr.append(c)
            p -= count[c] // 2
            if p <= 0:
                return 0

        arr.sort()
        diff =[]
        for i in range(len(arr)-1):
            diff.append(arr[i+1] - arr[i])

        diff.sort()
        return diff[p-1]


```

wrong




```python
class Solution:
    def minimizeMax(self, nums: List[int], p: int) -> int:
        nums.sort()
        n = len(nums)

        def greedyCount(threshold):
            index, count = 0, 0
            while index < n-1:
                if nums[index+1] - nums[index] <= threshold:
                    count += 1
                    index += 1
                index += 1
            return count

        l, r = 0, nums[-1]-nums[0]
        while l < r:
            mid = l + (r - l)// 2
            if greedyCount(mid) >= p:
                r = mid
            else:
                l = mid + 1

        return l
```

correct

greedy + binary search