Problem 2616: Minimize the Maximum Difference of Pairs

Problem Information

Difficulty: Medium
Acceptance Rate: 50.92%
Paid Only: No
Tags: Array, Binary Search, Dynamic Programming, Greedy, Sorting

Problem Description

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.

Example 2:

Input: nums = [4,2,1,2], p = 1 **Output:** 0 **Explanation:** Let the indices 1 and 3 form a pair. The difference of that pair is |2 - 2| = 0, which is the minimum we can attain.

Constraints:

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109` * `0 <= p <= (nums.length)/2`

Code Snippets

C++:

```
class Solution {
public:
int minimizeMax(vector<int>& nums, int p) {

}
};
```

Java:

```
class Solution {
public int minimizeMax(int[] nums, int p) {

}
}
```

Python3:

```
class Solution:
def minimizeMax(self, nums: List[int], p: int) -> int:
```