

## 2616. Minimize the Maximum Difference of Pairs

Hint

Medium 1.6K 149

Companies

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  $|\text{nums}[i] - \text{nums}[j]|$ , 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  $\max(|\text{nums}[1] - \text{nums}[4]|, |\text{nums}[2] - \text{nums}[5]|) = \max(0, 1) = 1$ . 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

i Java Auto

```
1 class Solution {
2     public boolean solve(int[] nums, int d, int p) {
3         int n = nums.length;
4         int c = 0;
5         for (int i = 0; i < n - 1; i++) {
6             if (nums[i + 1] - nums[i] <= d) {
7                 c++;
8                 i++;
9             }
10            if (c >= p) {
11                return true;
12            }
13        }
14        return false;
15    }
16
17    public int minimizeMax(int[] nums, int p) {
18        ...
19    }
20 }
```

Testcase Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums =  
[10,1,2,7,1,3]

p =  
2

Output

Console

Run

Submit

## 2544. Alternating Digit Sum

Hint

Easy 307 17

Companies

You are given a positive integer  $n$ . Each digit of  $n$  has a sign according to the following rules:

- The **most significant digit** is assigned a **positive** sign.
- Each other digit has an opposite sign to its adjacent digits.

Return the sum of all digits with their corresponding sign.

### Example 1:

**Input:**  $n = 521$

**Output:** 4

**Explanation:**  $(+5) + (-2) + (+1) = 4$ .

### Example 2:

**Input:**  $n = 111$

**Output:** 1

**Explanation:**  $(+1) + (-1) + (+1) = 1$ .

### Example 3:

**Input:**  $n = 886996$

**Output:** 0

**Explanation:**  $(+8) + (-8) + (+6) + (-9) + (+9) + (-6) = 0$ .

i Java Auto

```
1 class Solution {
2     public int alternateDigitSum(int n) {
3         String s = String.valueOf(n);
4         int factor = 1;
5         int res = 0;
6
7         for (char chr : s.toCharArray()) {
8             res += (Character.getNumericValue(chr) * factor);
9             factor *= -1;
10        }
11        return res;
12    }
13 }
```

Testcase Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

n =  
521

Output

4

Expected

Console



Run

Submit