Problem 719: Find K-th Smallest Pair Distance

Problem Information

Difficulty: Hard
Acceptance Rate: 46.13%
Paid Only: No
Tags: Array, Two Pointers, Binary Search, Sorting

Problem Description

The **distance of a pair** of integers `a` and `b` is defined as the absolute difference between `a` and `b`.

Given an integer array `nums` and an integer `k`, return _the_ `kth` _smallest**distance among all the pairs**_ `nums[i]` _and_ `nums[j]` _where_ `0 <= i < j < nums.length`.

Example 1:

Input: nums = [1,3,1], k = 1 **Output:** 0 **Explanation:** Here are all the pairs: (1,3) -> 2 (1,1) -> 0 (3,1) -> 2 Then the 1st smallest distance pair is (1,1), and its distance is 0.

Example 2:

Input: nums = [1,1,1], k = 2 **Output:** 0

Example 3:

Input: nums = [1,6,1], k = 3 **Output:** 5

Constraints:

* `n == nums.length` * `2 <= n <= 104` * `0 <= nums[i] <= 106` * `1 <= k <= n * (n - 1) / 2`

Code Snippets

C++:

```
class Solution {
public:
int smallestDistancePair(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int smallestDistancePair(int[] nums, int k) {

}
}
```

Python3:

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