Problem 2040: Kth Smallest Product of Two Sorted Arrays

Problem Information

Difficulty: Hard
Acceptance Rate: 48.99%
Paid Only: No
Tags: Array, Binary Search

Problem Description

Given two **sorted 0-indexed** integer arrays `nums1` and `nums2` as well as an integer `k`, return _the_`kth` _(**1-based**) smallest product of _`nums1[i] * nums2[j]`_where_`0 <= i < nums1.length` _and_`0 <= j < nums2.length`.

Example 1:

Input: nums1 = [2,5], nums2 = [3,4], k = 2 **Output:** 8 **Explanation:** The 2 smallest products are: - nums1[0] * nums2[0] = 2 * 3 = 6 - nums1[0] * nums2[1] = 2 * 4 = 8 The 2nd smallest product is 8.

Example 2:

Input: nums1 = [-4,-2,0,3], nums2 = [2,4], k = 6 **Output:** 0 **Explanation:** The 6 smallest products are: - nums1[0] * nums2[1] = (-4) * 4 = -16 - nums1[0] * nums2[0] = (-4) * 2 = -8 - nums1[1] * nums2[1] = (-2) * 4 = -8 - nums1[1] * nums2[0] = (-2) * 2 = -4 - nums1[2] * nums2[0] = 0 * 2 = 0 - nums1[2] * nums2[1] = 0 * 4 = 0 The 6th smallest product is 0.

Example 3:

Input: nums1 = [-2,-1,0,1,2], nums2 = [-3,-1,2,4,5], k = 3 **Output:** -6 **Explanation:** The 3 smallest products are: - nums1[0] * nums2[4] = (-2) * 5 = -10 - nums1[0] * nums2[3] = (-2) * 4 = -8 - nums1[4] * nums2[0] = 2 * (-3) = -6 The 3rd smallest product is -6.

Constraints:

* `1 <= nums1.length, nums2.length <= 5 * 104` * `-105 <= nums1[i], nums2[j] <= 105` * `1 <= k <= nums1.length * nums2.length` * `nums1` and `nums2` are sorted.

Code Snippets

C++:

```
class Solution {
public:
long long kthSmallestProduct(vector<int>& nums1, vector<int>& nums2, long
long k) {

}
};
```

Java:

```
class Solution {
public long kthSmallestProduct(int[] nums1, int[] nums2, long k) {

}
}
```

Python3:

```
class Solution:
def kthSmallestProduct(self, nums1: List[int], nums2: List[int], k: int) ->
int:
```