5887. Kth Smallest Product of Two Sorted Arrays

My Submissions (/contest/biweekly-contest-63/problems/kth-smallest-product-of-two-sorted-arrays/submissions/)

Back to Contest (/contest/biweekly-contest-63/)

Given two **sorted 0-indexed** integer arrays `nums1` and `nums2` as well as an integer `k`, return *the* k^{th} *(1-based) smallest product of* `nums1[i] * nums2[j]` *where* `0 <= i < nums1.length` *and* `0 <= j < nums2.length`.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

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 * 10^4`
- `-10^5 <= nums1[i], nums2[j] <= 10^5`
- `1 <= k <= nums1.length * nums2.length`
- `nums1` and `nums2` are sorted.

JavaScript ▾   ⟨⟩ ⟳ ⚙

```javascript
1    const ll = BigInt;
2
3 ▾  function Bisect() {
4        return { insort_right, insort_left, bisect_left, bisect_right }
5 ▾      function insort_right(a, x, lo = 0, hi = null) {
6            lo = bisect_right(a, x, lo, hi);
7            a.splice(lo, 0, x);
8        }
9 ▾      function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
10           if (lo < 0) throw new Error('lo must be non-negative');
```

```
11          if (hi == null) hi = a.length;
12 ▾        while (lo < hi) {
13              let mid = lo + hi >> 1;
14              x < a[mid] ? hi = mid : lo = mid + 1;
15          }
16          return lo;
17      }
18 ▾    function insort_left(a, x, lo = 0, hi = null) {
19          lo = bisect_left(a, x, lo, hi);
20          a.splice(lo, 0, x);
21      }
22 ▾    function bisect_left(a, x, lo = 0, hi = null) { // >= lower_bound
23          if (lo < 0) throw new Error('lo must be non-negative');
24          if (hi == null) hi = a.length;
25 ▾        while (lo < hi) {
26              let mid = parseInt((lo + hi) / 2);
27              a[mid] < x ? lo = mid + 1 : hi = mid;
28          }
29          return lo;
30      }
31  }
32
33
34  const floorDiv = (x, d) => { let r = parseInt(x / d); if (r * d != x && (x ^ d) < 0) r--; return r; };
35  const floorDivll = (x, d) => { let r = x / d; if (r * d != x && (x ^ d) < 0) r--; return r; };
36
37 ▾ const kthSmallestProduct = (a, b, k) => {
38      let bi = new Bisect(), bn = b.length;
39      let low = -2e10, high = 2e10;
40 ▾    while (high - low > 1) {
41          let m = parseInt((low + high) / 2);
42          // pr(low, high, m);
43          let cnt = 0;
44 ▾        for (const x of a) {
45 ▾            if (x > 0) {
46                  let t = floorDivll(ll(m), ll(x));
47                  // let t = floorDiv(m, x);
48                  let idx = bi.bisect_left(b, Number(t) + 1);
49                  cnt += idx;
50 ▾            } else if (x < 0) {
51                  let t = floorDivll(ll(m) + ll(x) + 1n, ll(x))
52                  // let t = floorDiv(m + x + 1, x);
53                  let idx = bi.bisect_left(b, Number(t));
54                  cnt += bn - idx;
55 ▾            } else {
56                  if (m >= 0) cnt += bn;
57              }
58          }
59          cnt >= k ? high = m : low = m;
60      }
61      // pr(low, high);
62      return high;
63  };
```

☐ **Custom Testcase**   (Use Example Testcases)

▶ Run   ☁ Submit

Submission Result: Accepted (/submissions/detail/572242274/) ❓   (More Details ❯ (/submissions/detail/572242274/))

Share your acceptance!