2542. Maximum Subsequence Score

Description

You are given two **0-indexed** integer arrays `nums1` and `nums2` of equal length `n` and a positive integer `k`. You must choose a **subsequence** of indices from `nums1` of length `k`.

For chosen indices i_0, i_1, ..., i_{k-1}, your **score** is defined as:

- The sum of the selected elements from `nums1` multiplied with the **minimum** of the selected elements from `nums2`.
- It can defined simply as: $(nums1[i_0] + nums1[i_1] + ... + nums1[i_{k-1}]) * \min(nums2[i_0], nums2[i_1], ..., nums2[i_{k-1}])$.

Return *the **maximum** possible score.*

A **subsequence** of indices of an array is a set that can be derived from the set `{0, 1, ..., n-1}` by deleting some or no elements.

Example 1:

```
Input: nums1 = [1,3,3,2], nums2 = [2,1,3,4], k = 3
Output: 12
Explanation:
The four possible subsequence scores are:
- We choose the indices 0, 1, and 2 with score = (1+3+3) * min(2,1,3) = 7.
- We choose the indices 0, 1, and 3 with score = (1+3+2) * min(2,1,4) = 6.
- We choose the indices 0, 2, and 3 with score = (1+3+2) * min(2,3,4) = 12.
- We choose the indices 1, 2, and 3 with score = (3+3+2) * min(1,3,4) = 8.
Therefore, we return the max score, which is 12.
```

Example 2:

```
Input: nums1 = [4,2,3,1,1], nums2 = [7,5,10,9,6], k = 1
Output: 30
Explanation:
Choosing index 2 is optimal: nums1[2] * nums2[2] = 3 * 10 = 30 is the maximum possible score.
```

Constraints:

- `n == nums1.length == nums2.length`
- `1 <= n <= 10^5`
- `0 <= nums1[i], nums2[j] <= 10^5`
- `1 <= k <= n`