You are given two integers n and k and two integer arrays speed and efficiency both of length n. There are n engineers numbered from 1 to n. speed[i] and efficiency[i] represent the speed and efficiency of the ith engineer respectively.

Choose **at most** k different engineers out of the n engineers to form a team with the maximum **performance**.

The performance of a team is the sum of their engineers' speeds multiplied by the minimum efficiency among their engineers.

Return *the maximum performance of this team*. Since the answer can be a huge number, return it **modulo** 109 + 7.

**Example 1:**

**Input:** n = 6, speed = [2,10,3,1,5,8], efficiency = [5,4,3,9,7,2], k = 2

**Output:** 60

**Explanation:**

We have the maximum performance of the team by selecting engineer 2 (with speed=10 and efficiency=4) and engineer 5 (with speed=5 and efficiency=7). That is, performance = (10 + 5) \* min(4, 7) = 60.

**Example 2:**

**Input:** n = 6, speed = [2,10,3,1,5,8], efficiency = [5,4,3,9,7,2], k = 3

**Output:** 68

**Explanation:**

This is the same example as the first but k = 3. We can select engineer 1, engineer 2 and engineer 5 to get the maximum performance of the team. That is, performance = (2 + 10 + 5) \* min(5, 4, 7) = 68.

**Example 3:**

**Input:** n = 6, speed = [2,10,3,1,5,8], efficiency = [5,4,3,9,7,2], k = 4

**Output:** 72

**Constraints:**

* 1 <= <= k <= n <= 105
* speed.length == n
* efficiency.length == n
* 1 <= speed[i] <= 105
* 1 <= efficiency[i] <= 108