# 2542. Maximum Subsequence Score

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 i0, i1, ..., ik - 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[i0] + nums1[i1] +...+ nums1[ik - 1]) \* min(nums2[i0] , nums2[i1], ... ,nums2[ik - 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.

## SOLUTION IN C++

class Solution {

public:

long long maxScore(vector<int>& nums1, vector<int>& nums2, int k) {

long ans = 0;

long sum = 0;

vector<pair<int, int>> A;

priority\_queue<int, vector<int>, greater<>> minHeap;

for (int i = 0; i < nums1.size(); ++i)

A.emplace\_back(nums2[i], nums1[i]);

ranges::sort(A, greater<>());

for (const auto& [num2, num1] : A) {

minHeap.push(num1);

sum += num1;

if (minHeap.size() > k)

sum -= minHeap.top(), minHeap.pop();

if (minHeap.size() == k)

ans = max(ans, sum \* num2);

}

return ans;

}

};