Given two sparse vectors, compute their dot product.

Implement class SparseVector:

* SparseVector(nums) Initializes the object with the vector nums
* dotProduct(vec) Compute the dot product between the instance of *SparseVector* and vec

A **sparse vector** is a vector that has mostly zero values, you should store the sparse vector **efficiently**and compute the dot product between two *SparseVector*.

**Follow up:**What if only one of the vectors is sparse?

**Example 1:**

**Input:** nums1 = [1,0,0,2,3], nums2 = [0,3,0,4,0]

**Output:** 8

**Explanation:** v1 = SparseVector(nums1) , v2 = SparseVector(nums2)

v1.dotProduct(v2) = 1\*0 + 0\*3 + 0\*0 + 2\*4 + 3\*0 = 8

**Example 2:**

**Input:** nums1 = [0,1,0,0,0], nums2 = [0,0,0,0,2]

**Output:** 0

**Explanation:** v1 = SparseVector(nums1) , v2 = SparseVector(nums2)

v1.dotProduct(v2) = 0\*0 + 1\*0 + 0\*0 + 0\*0 + 0\*2 = 0

**Example 3:**

**Input:** nums1 = [0,1,0,0,2,0,0], nums2 = [1,0,0,0,3,0,4]

**Output:** 6

**Constraints:**

* n == nums1.length == nums2.length
* 1 <= n <= 10^5
* 0 <= nums1[i], nums2[i] <= 100