Problem 1570: Dot Product of Two Sparse Vectors

Problem Information

Difficulty: Medium
Acceptance Rate: 89.94%
Paid Only: Yes
Tags: Array, Hash Table, Two Pointers, Design

Problem Description

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`

Code Snippets

C++:

```
class SparseVector {
public:

SparseVector(vector<int> &nums) {

}

// Return the dotProduct of two sparse vectors
int dotProduct(SparseVector& vec) {

}
};

// Your SparseVector object will be instantiated and called as such:
// SparseVector v1(nums1);
// SparseVector v2(nums2);
// int ans = v1.dotProduct(v2);
```

Java:

```
class SparseVector {

SparseVector(int[] nums) {

}

// Return the dotProduct of two sparse vectors
public int dotProduct(SparseVector vec) {
}
}

// Your SparseVector object will be instantiated and called as such:
// SparseVector v1 = new SparseVector(nums1);
// SparseVector v2 = new SparseVector(nums2);
// int ans = v1.dotProduct(v2);
```

Python3:

```
class SparseVector:
def __init__(self, nums: List[int]):



# Return the dotProduct of two sparse vectors
def dotProduct(self, vec: 'SparseVector') -> int:



# Your SparseVector object will be instantiated and called as such:
# v1 = SparseVector(nums1)
# v2 = SparseVector(nums2)
# ans = v1.dotProduct(v2)
```