Given a nested list of integers, return the sum of all integers in the list weighted by their depth.

Each element is either an integer, or a list -- whose elements may also be integers or other lists.

**Example 1:**  
Given the list [[1,1],2,[1,1]], return **10**. (four 1's at depth 2, one 2 at depth 1)

**Example 2:**  
Given the list [1,[4,[6]]], return **27**. (one 1 at depth 1, one 4 at depth 2, and one 6 at depth 3; 1 + 4\*2 + 6\*3 = 27)

**Solution:** Go with the basic intuition that this has to be solved by recursion.

* If an element is just an integer, then add it to the running sum.
* If an element is a list, then recursively call the same function to get the sum of the sublist. Add it to the running sum.
* Return the sum.