LeetCode Explore Problems Interview New Contest Discuss Store

LeetCoding Challenge + GIVEAWAY!

Description Solution Discuss (469) Submissions

364. Nested List Weight Sum II

Medium 840 245 Add to List Share

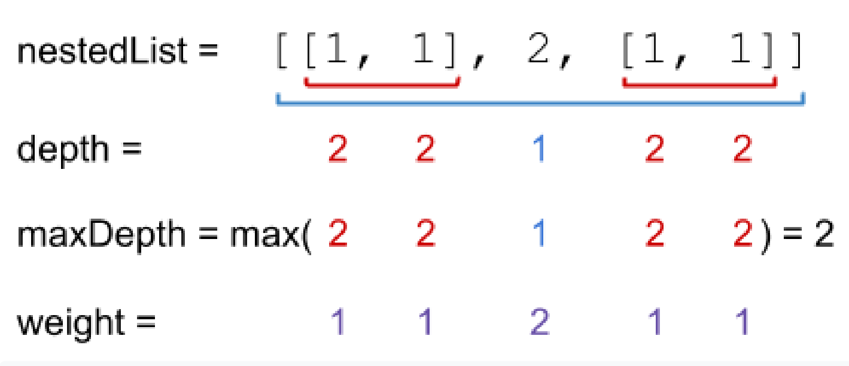
You are given a nested list of integers `nestedList`. Each element is either an integer or a list whose elements may also be integers or other lists.

The **depth** of an integer is the number of lists that it is inside of. For example, the nested list `[1,[2,2],[[3],2],1]` has each integer's value set to its **depth**. Let `maxDepth` be the **maximum depth** of any integer.

The **weight** of an integer is `maxDepth - (the depth of the integer) + 1`.

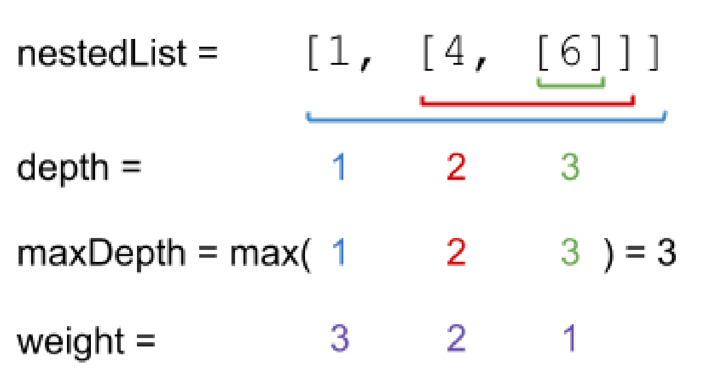
Return *the sum of each integer in* `nestedList` *multiplied by its* ***weight***.

Example 1:


nestedList = [[1, 1], 2, [1, 1]]
depth = 2 2 1 2 2
maxDepth = max(2 2 1 2 2) = 2
weight = 1 1 2 1 1

```
Input: nestedList = [[1,1],2,[1,1]]
Output: 8
Explanation: Four 1's with a weight of 1, one 2 with a weight of 2.
1*1 + 1*1 + 2*2 + 1*1 + 1*1 = 8
```

Example 2:



```
Input: nestedList = [1,[4,[6]]]
Output: 17
Explanation: One 1 at depth 3, one 4 at depth 2, and one 6 at depth 1.
1*3 + 4*2 + 6*1 = 17
```

Constraints:

- `1 <= nestedList.length <= 50`
- The values of the integers in the nested list is in the range `[-100, 100]`.
- The maximum **depth** of any integer is less than or equal to `50`.

Accepted 95,216 Submissions 143,538

Seen this question in a real interview before? Yes No

Companies *i*

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

LinkedIn 35

Related Topics

Stack Depth-First Search Breadth-First Search

Similar Questions

Nested List Weight Sum	Medium
Array Nesting	Medium

Java Autocomplete

```java
/**
 * // This is the interface that allows for creating nested lists.
 * // You should not implement it, or speculate about its implementation
 * public interface NestedInteger {
 *     // Constructor initializes an empty nested list.
 *     public NestedInteger();
 *
 *     // Constructor initializes a single integer.
 *     public NestedInteger(int value);
 *
 *     // @return true if this NestedInteger holds a single integer, rather than a nested list.
 *     public boolean isInteger();
 *
 *     // @return the single integer that this NestedInteger holds, if it holds a single integer
 *     // Return null if this NestedInteger holds a nested list
 *     public Integer getInteger();
 *
 *     // Set this NestedInteger to hold a single integer.
 *     public void setInteger(int value);
 *
 *     // Set this NestedInteger to hold a nested list and adds a nested integer to it.
 *     public void add(NestedInteger ni);
 *
 *     // @return the nested list that this NestedInteger holds, if it holds a nested list
 *     // Return empty list if this NestedInteger holds a single integer
 *     public List<NestedInteger> getList();
 * }
 */
class Solution {
    public int depthSumInverse(List<NestedInteger> nestedList) {
        Queue<NestedInteger> Q = new LinkedList<>();
        Q.addAll(nestedList);

        int depth = 1;
        int maxDepth = 0;
        int sumOfElements = 0;
        int sumOfProducts = 0;

        while (!Q.isEmpty()) {
            int size = Q.size();
            maxDepth = Math.max(maxDepth, depth);

            for (int i = 0; i < size; i++) {
                NestedInteger nested = Q.poll();

                if (nested.isInteger()) {
                    sumOfElements += nested.getInteger();
                    sumOfProducts += nested.getInteger() * depth;
                } else {
                    Q.addAll(nested.getList());
                }
            }
            depth++;
        }
        return (maxDepth + 1) * sumOfElements - sumOfProducts;
    }
}
```

Problems Pick One Prev /99 Next Console Contribute *i* Run Code Submit