726. Number of Atoms

- Difficulty:Hard
- Total Accepted:1.6K
- Total Submissions:3.4K
- Contributor: 1337c0d3r
-

- Subscribe to see which companies asked this question.

Related Topics String Recursive

Similar Questions

Decode String Encode String with Shortest Length

Given a chemical `formula` (given as a string), return the count of each atom.

An atomic element always starts with an uppercase character, then zero or more lowercase letters, representing the name.

1 or more digits representing the count of that element may follow if the count is greater than 1. If the count is 1, no digits will follow. For example, H2O and H2O2 are possible, but H1O2 is impossible.

Two formulas concatenated together produce another formula. For example, H2O2He3Mg4 is also a formula.

A formula placed in parentheses, and a count (optionally added) is also a formula. For example, (H2O2) and (H2O2)3 are formulas.

Given a formula, output the count of all elements as a string in the following form: the first name (in sorted order), followed by its count (if that count is more than 1), followed by the second name (in sorted order), followed by its count (if that count is more than 1), and so on.

Example 1:

```
Input:
formula = "H2O"
Output: "H2O"
Explanation:
The count of elements are {'H': 2, 'O': 1}.
```

Example 2:

```
Input:
formula = "Mg(OH)2"
Output: "H2MgO2"
Explanation:
The count of elements are {'H': 2, 'Mg': 1, 'O': 2}.
```

Example 3:

Input:
formula = "K4(ON(SO3)2)2"
Output: "K4N2O14S4"
Explanation:
The count of elements are {'K': 4, 'N': 2, 'O': 14, 'S': 4}.

Note:

- All atom names consist of lowercase letters, except for the first character which is uppercase.
- The length of `formula` will be in the range `[1, 1000]`.
- `formula` will only consist of letters, digits, and round parentheses, and is a valid formula as defined in the problem.

```
def countOfAtoms(self, formula):
        # read a number
        def read_num(formula, i):
            num = 0
            while i < len(formula) and formula[i].isdigit():
                num = 10 * num + int(formula[i])
                i += 1
            if num == 0:
                num = 1
            return num, i

        # read an Atom, a upper letter followed by several  lower case letters
        def read_atom(formula, i):
            atom = formula[i]
            i += 1
            while i < len(formula) and formula[i].islower():
                atom += formula[i]
                i += 1
            return atom, i

        # read the formula inside parentheses
        def read_formula(formula, i):
            from collections import defaultdict
            atoms = defaultdict(int)
            while i < len(formula) and formula[i] != ')':
                if formula[i] == '(':
                    child_atoms, i = read_formula(formula, i + 1)
                    num, i = read_num(formula, i)
                    for k, v in child_atoms.items():
                        atoms[k] += v*num
                else:
                    atom, i = read_atom(formula, i)
                    num, i = read_num(formula, i)
                    atoms[atom] += num
            return atoms, i + 1

        i = 0
        atoms, i = read_formula(formula, i)
        result = ''
        for atom, num in sorted(atoms.items()):
            result += atom
            if num > 1:
```

```
        result += str(num)
return result
```