811. Subdomain Visit Count

A website domain `"discuss.leetcode.com"` consists of various subdomains. At the top level, we have `"com"`, at the next level, we have `"leetcode.com"` and at the lowest level, `"discuss.leetcode.com"`. When we visit a domain like `"discuss.leetcode.com"`, we will also visit the parent domains `"leetcode.com"` and `"com"` implicitly.

A **count-paired domain** is a domain that has one of the two formats `"rep d1.d2.d3"` or `"rep d1.d2"` where `rep` is the number of visits to the domain and `d1.d2.d3` is the domain itself.

- For example, `"9001 discuss.leetcode.com"` is a **count-paired domain** that indicates that `discuss.leetcode.com` was visited `9001` times.

Given an array of **count-paired domains** `cpdomains`, return *an array of the **count-paired domains** of each subdomain in the input*. You may return the answer in **any order**.

Example 1:

```
Input: cpdomains = ["9001 discuss.leetcode.com"]
Output: ["9001 leetcode.com","9001 discuss.leetcode.com","9001 com"]
Explanation: We only have one website domain: "discuss.leetcode.com".
As discussed above, the subdomain "leetcode.com" and "com" will also be
visited. So they will all be visited 9001 times.
```

Example 2:

```
Input: cpdomains = ["900 google.mail.com", "50 yahoo.com", "1
intel.mail.com", "5 wiki.org"]
Output: ["901 mail.com","50 yahoo.com","900 google.mail.com","5 wiki.org","5
org","1 intel.mail.com","951 com"]
Explanation: We will visit "google.mail.com" 900 times, "yahoo.com" 50
times, "intel.mail.com" once and "wiki.org" 5 times.
For the subdomains, we will visit "mail.com" 900 + 1 = 901 times, "com" 900
+ 50 + 1 = 951 times, and "org" 5 times.
```

Constraints:

- `1 <= cpdomain.length <= 100`
- `1 <= cpdomain[i].length <= 100`
- `cpdomain[i]` follows either the `"rep<sub>i</sub> d1<sub>i</sub>.d2<sub>i</sub>.d3<sub>i</sub>"` format or the `"rep<sub>i</sub> d1<sub>i</sub>.d2<sub>i</sub>"` format.

- `rep<sub>i</sub>` is an integer in the range `[1, 10<sup>4</sup>]`.
- `d1<sub>i</sub>`, `d2<sub>i</sub>`, and `d3<sub>i</sub>` consist of lowercase English letters.

```
class Solution:
    def subdomainVisits(self, cpdomains: List[str]) -> List[str]:
        freq = {}
        ans = []
        for ele in cpdomains:
            arr = self.returnFixedString(ele)
            adder = arr[0]
            for i in range(1, len(arr)):
                if arr[i] in freq:
                    freq[arr[i]] = freq[arr[i]] + adder
                else:
                    freq[arr[i]] = adder

        for key, val in freq.items():
            temp = str(val) + ' ' + key
            ans.append(temp)
        return ans

    def returnFixedString(self,string):
        string = string.replace(' ', '.').split('.')
        # string = [string[0]]+ ['.' + string[i] for i in
range(1,len(string))]
        ans = [None] * len(string)
        temp = string[-1]
        ans[-1] = temp
        for i in range(len(string) - 2, 0, -1):
            temp = string[i] + '.' + temp
            ans[i] = temp
        ans[0] = int(string[0])

        return ans
```