

1487. Making File Names Unique

Description

Given an array of strings `names` of size `n`. You will create `n` folders in your file system **such that**, at the `ith` minute, you will create a folder with the name `names[i]`.

Since two files **cannot** have the same name, if you enter a folder name that was previously used, the system will have a suffix addition to its name in the form of `(k)`, where, `k` is the **smallest positive integer** such that the obtained name remains unique.

Return *an array of strings of length* `n` where `ans[i]` is the actual name the system will assign to the `ith` folder when you create it.

Example 1:

```
Input: names = ["pes","fifa","gta","pes(2019)"]
Output: ["pes","fifa","gta","pes(2019)"]
Explanation: Let's see how the file system creates folder names:
"pes" --> not assigned before, remains "pes"
"fifa" --> not assigned before, remains "fifa"
"gta" --> not assigned before, remains "gta"
"pes(2019)" --> not assigned before, remains "pes(2019)"
```

Example 2:

```
Input: names = ["gta","gta(1)","gta","avalon"]
Output: ["gta","gta(1)","gta(2)","avalon"]
Explanation: Let's see how the file system creates folder names:
"gta" --> not assigned before, remains "gta"
"gta(1)" --> not assigned before, remains "gta(1)"
"gta" --> the name is reserved, system adds (k), since "gta(1)" is also reserved, systems put k = 2. it becomes "gta(2)"
"avalon" --> not assigned before, remains "avalon"
```

Example 3:

```
Input: names = ["onepiece","onepiece(1)","onepiece(2)","onepiece(3)","onepiece"]
Output: ["onepiece","onepiece(1)","onepiece(2)","onepiece(3)","onepiece(4)"]
Explanation: When the last folder is created, the smallest positive valid k is 4, and it becomes "onepiece(4)".
```

Constraints:

- `1 <= names.length <= 5 * 104`
- `1 <= names[i].length <= 20`
- `names[i]` consists of lowercase English letters, digits, and/or round brackets.

