Given a list of folders, remove all sub-folders in those folders and return in **any order** the folders after removing.

If a folder[i] is located within another folder[j], it is called a sub-folder of it.

The format of a path is one or more concatenated strings of the form: / followed by one or more lowercase English letters. For example, /leetcode and /leetcode/problems are valid paths while an empty string and / are not.

**Example 1:**

**Input:** folder = ["/a","/a/b","/c/d","/c/d/e","/c/f"]

**Output:** ["/a","/c/d","/c/f"]

**Explanation:** Folders "/a/b/" is a subfolder of "/a" and "/c/d/e" is inside of folder "/c/d" in our filesystem.

**Example 2:**

**Input:** folder = ["/a","/a/b/c","/a/b/d"]

**Output:** ["/a"]

**Explanation:** Folders "/a/b/c" and "/a/b/d/" will be removed because they are subfolders of "/a".

**Example 3:**

**Input:** folder = ["/a/b/c","/a/b/ca","/a/b/d"]

**Output:** ["/a/b/c","/a/b/ca","/a/b/d"]

**Constraints:**

* 1 <= folder.length <= 4 \* 10^4
* 2 <= folder[i].length <= 100
* folder[i] contains only lowercase letters and '/'
* folder[i] always starts with character '/'
* Each folder name is unique.