# Problem 1233: Remove Sub-Folders from the Filesystem

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 78.63%
**Paid Only:** No
**Tags:** Array, String, Depth-First Search, Trie

## Problem Description

Given a list of folders `folder`, return _the folders after removing all**sub- folders** in those folders_. You may return the answer in **any order**.

If a `folder[i]` is located within another `folder[j]`, it is called a **sub- folder** of it. A sub-folder of `folder[j]` must start with `folder[j]`, followed by a `"/"`. For example, `"/a/b"` is a sub-folder of `"/a"`, but `"/b"` is not a sub-folder of `"/a/b/c"`.

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 * 104` * `2 <= folder[i].length <= 100` * `folder[i]` contains only lowercase letters and `'/'`. * `folder[i]` always starts with the character `'/'`. * Each folder name is **unique**.

## Code Snippets

**C++:**

```
class Solution {
public:
vector<string> removeSubfolders(vector<string>& folder) {

}
};
```

**Java:**

```
class Solution {
public List<String> removeSubfolders(String[] folder) {

}
}
```

**Python3:**

```
class Solution:
def removeSubfolders(self, folder: List[str]) -> List[str]:
```