Remove Sub Folders From The Filesystem

solved by	Senan
Platform	LeetCode
difficulty	Medium
Serial	1233
tags	Sorting String Manipulation Vector
language	C++
solved on	@25/10/2024
link	https://leetcode.com/problems/remove-sub-folders-from-the-filesystem/description/
Completion	✓

Intuition

The task is to remove subfolders from a list of folder paths while keeping only the top-level folders. A subfolder is a folder that resides within a parent folder, identifiable by the `/` symbol in the path. Sorting the folders lexicographically ensures that subfolders appear directly after their parent folders, simplifying the removal process.

Approach

1. **Sorting**: First, sort the folder paths lexicographically so that subfolders appear right after their parent folders.
2. **Helper Function (`isSubFolder`)**: Define a helper function to check if one folder is a subfolder of another by comparing their characters and ensuring the subfolder starts with the parent folder and is followed by a `/`.
3. **Iterating**: After sorting, iterate through the list of folders. For each folder, check if it is a subfolder of the last added folder. If it's not, add it to the result list.

Complexity

Time Complexity:

- Sorting the folders takes **O(n log n)**, where **n** is the number of folders.
- Comparing folder paths in the helper function takes **O(L)**, where **L** is the maximum length of a folder path.
- The overall time complexity is **O(n log n + nL)**.

Space Complexity:

- The space complexity is **O(n)** for storing the result.

Code

```
class Solution {
    bool isSubFolder(const string& s1, const string& s2) {
        int n1 = s1.size(), n2 = s2.size();
        if (n1 >= n2) return false;
        int i = 0, j = 0;
        while (i < n1) {
            if (s1[i++] != s2[j++]) return false;
        }
        return s2[j] == '/';
    }
public:
    vector<std::string> removeSubfolders(vector<string>& folder) {
        vector<string> answer;
        sort(folder.begin(), folder.end());
        answer.push_back(folder[0]);
        for (int i = 1; i < folder.size(); i++) {
            if (!isSubFolder(answer.back(), folder[i])) {
                answer.push_back(folder[i]);
            }
        }
        return answer;
    }
};
```