

100078. Longest Unequal Adjacent Groups Subsequence I

My Submissions (/contest/biweekly-contest-115/problems/longest-unequal-adjacent-groups-subsequence-i/submissions/)

Back to Contest (/contest/biweekly-contest-115/)

You are given an integer n , a **0-indexed** string array `words`, and a **0-indexed binary** array `groups`, both arrays having length n .

You need to select the **longest subsequence** from an array of indices $[0, 1, \dots, n - 1]$, such that for the subsequence denoted as $[i_0, i_1, \dots, i_{k-1}]$ having length k , $groups[i_j] \neq groups[i_{j+1}]$, for each j where $0 \leq j + 1 < k$.

Return a string array containing the words corresponding to the indices (**in order**) in the selected subsequence. If there are multiple answers, return *any of them*.

A **subsequence** of an array is a new array that is formed from the original array by deleting some (possibly none) of the elements without disturbing the relative positions of the remaining elements.

Note: strings in `words` may be **unequal** in length.

User Accepted:	701
User Tried:	786
Total Accepted:	708
Total Submissions:	859
Difficulty:	Medium

Example 1:

Input: `n = 3, words = ["e","a","b"], groups = [0,0,1]`
Output: `["e","b"]`
Explanation: A subsequence that can be selected is `[0,2]` because `groups[0] != groups[2]`. So, a valid answer is `[words[0],words[2]] = ["e","b"]`. Another subsequence that can be selected is `[1,2]` because `groups[1] != groups[2]`. This results in `[words[1],words[2]] = ["a","b"]`. It is also a valid answer. It can be shown that the length of the longest subsequence of indices that satisfies the condition is 2.

Example 2:

Input: `n = 4, words = ["a","b","c","d"], groups = [1,0,1,1]`
Output: `["a","b","c"]`
Explanation: A subsequence that can be selected is `[0,1,2]` because `groups[0] != groups[1]` and `groups[1] != groups[2]`. So, a valid answer is `[words[0],words[1],words[2]] = ["a","b","c"]`. Another subsequence that can be selected is `[0,1,3]` because `groups[0] != groups[1]` and `groups[1] != groups[3]`. This results in `[words[0],words[1],words[3]] = ["a","b","d"]`. It is also a valid answer. It can be shown that the length of the longest subsequence of indices that satisfies the condition is 3.

Constraints:

- $1 \leq n \leq words.length \leq groups.length \leq 100$
- $1 \leq words[i].length \leq 10$
- $0 \leq groups[i] < 2$
- `words` consists of **distinct** strings.
- `words[i]` consists of lowercase English letters.

JavaScript

📄

↺

⚙️


```
1 const getWordsInLongestSubsequence = (n, a, b) => {
2   let d = [], res = [];
3   for (let i = 0; i < n; i++) {
4     let s = [b[i]]; indice = [i];
5     for (let j = i + 1; j < n; j++) {
6       let last = s[s.length - 1], expect = last ^ 1;
7       if (b[j] == expect) {
8         s.push(b[j]);
9         indice.push(j);
10      }
11    }
12    d.push([s, indice])
13  }
```

```
14     d.sort((x, y) => y[1].length - x[1].length)
15     for (const idx of d[0][1]) res.push(a[idx]);
16     return res;
17 };
```

☐ Custom Testcase☒ Use Example TestcasesSubmission Result: **Accepted** (/submissions/detail/1075071754/) ⓘ[More Details >](/submissions/detail/1075071754/)

Share your acceptance!

Copyright © 2023 LeetCode

[Help Center \(/support/\)](/support/) | [Jobs \(/jobs/\)](/jobs/) | [Bug Bounty \(/bugbounty/\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student/\)](/student/) | [Terms \(/terms/\)](/terms/) | [Privacy Policy \(/privacy/\)](/privacy/) [United States \(/region/\)](/region/)