# 6924. Length of the Longest Valid Substring

My Submissions (/contest/weekly-contest-354/problems/length-of-the-longest-valid-substring/submissions/) Back to Contest (/contest/weekly-contest-354/)

| | |
|---|---|
| User Accepted: | 0 |
| User Tried: | 0 |
| Total Accepted: | 0 |
| Total Submissions: | 0 |
| Difficulty: | Hard |

You are given a string `word` and an array of strings `forbidden` .

A string is called **valid** if none of its substrings are present in `forbidden` .

Return *the length of the **longest valid substring** of the string* `word` .

A **substring** is a contiguous sequence of characters in a string, possibly empty.

**Example 1:**

```
Input: word = "cbaaaabc", forbidden = ["aaa","cb"]
Output: 4
Explanation: There are 9 valid substrings in word: "c", "b", "a", "ba", "aa", "bc", "baa", "aab", and "aabc". The length of the
It can be shown that all other substrings contain either "aaa" or "cb" as a substring.
```

**Example 2:**

```
Input: word = "leetcode", forbidden = ["de","le","e"]
Output: 4
Explanation: There are 11 valid substrings in word: "l", "t", "c", "o", "d", "tc", "co", "od", "tco", "cod", and "tcod". The le
It can be shown that all other substrings contain either "de", "le", or "e" as a substring.
```

**Constraints:**

- `1 <= word.length <= 10`$^5$
- `word` consists only of lowercase English letters.
- `1 <= forbidden.length <= 10`$^5$
- `1 <= forbidden[i].length <= 10`
- `forbidden[i]` consists only of lowercase English letters.

Java

```java
class Solution {
    public int longestValidSubstring(String word, List<String> forbidden) {
        int n = word.length(), l = 0, res = 0;
        Trie tree = new Trie();
        for (String s : forbidden) tree.insert(s);
        for (int i = 0; i < n; i++) {
            while (tree.isValid(word, l, i)) l++;
            res = Math.max(res, i - l + 1);
        }
        return res;
    }

    class Trie {
        Trie[] next;
        int cnt;
        boolean end;

        Trie() {
            this.next = new Trie[26];
            this.cnt = 0;
            this.end = false;
        }

        void insert(String s) {
            Trie cur = this;
            for (int i = s.length() - 1; i >= 0; i--) {
                int idx = s.charAt(i) - 'a';
                if (cur.next[idx] == null) cur.next[idx] = new Trie();
                cur = cur.next[idx];
                cur.cnt++;
            }
            cur.end = true;
        }

        boolean isValid(String s, int l, int r) {
            Trie cur = this;
            for (int i = r; i >= l; i--) {
                int idx = s.charAt(i) - 'a';
                if (cur.next[idx] == null) return false;
                cur = cur.next[idx];
                if (cur.end) return true;
            }
            return false;
        }
    }
}
```

☐ **Custom Testcase** Use Example Testcases

Run Submit

**Submission Result:** Accepted (/submissions/detail/998909923/) More Details > (/submissions/detail/998909923/)

Share your acceptance!

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)