

748. Shortest Completing Word

Solved ✓

Easy

Topics

Companies

Hint

Given a string `licensePlate` and an array of strings `words`, find the **shortest completing** word in `words`.

A **completing** word is a word that **contains all the letters** in `licensePlate`. **Ignore numbers and spaces** in `licensePlate`, and treat letters as **case insensitive**. If a letter appears more than once in `licensePlate`, then it must appear in the word the same number of times or more.

For example, if `licensePlate` = "aBc 12c", then it contains letters 'a', 'b' (ignoring case), and 'c' twice. Possible **completing** words are "abccdef", "caaacab", and "cbca".

Return the **shortest completing** word in `words`. It is guaranteed an answer exists. If there are multiple shortest **completing** words, return the **first** one that occurs in `words`.

Example 1:

Input: `licensePlate` = "1s3 PSt", `words` = ["step","steps","stripe","stepple"]

Output: "steps"

Explanation: `licensePlate` contains letters 's', 'p', 's' (ignoring case), and 't'.

"step" contains 't' and 'p', but only contains 1 's'.

"steps" contains 't', 'p', and both 's' characters.

"stripe" is missing an 's'.

"stepple" is missing an 's'.

Since "steps" is the only word containing all the letters, that is the answer.

The screenshot shows a coding platform interface with the following components:

- Problem List:** A tab at the top left showing the current problem.
- Description:** A tab showing the problem details, including the input/output and explanation.
- Accepted:** A green badge indicating the solution is accepted.
- Submission:** A green button labeled "Solution" next to the submission details.
- Performance:** A bar chart showing the runtime of the solution across different test cases. The chart shows a peak at 90ms.
- Code:** A code editor showing the C# solution. The code is as follows:

```
C#  
using System;  
using System.Collections.Generic;  
using System.Linq;  
  
public class Solution {  
    public string ShortestCompletingWord(string licensePlate, string[] words)  
    {  
        // ... (code omitted for brevity) ...  
    }  
}
```

Код:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
  
public class Solution  
{  
    public string ShortestCompletingWord(string licensePlate, string[] words)
```

```

{
    // Создаем словарь для подсчета количества каждой буквы в licensePlate
    Dictionary<char, int> targetCount = new Dictionary<char, int>();

    foreach (char c in licensePlate.ToLower())
    {
        if (char.IsLetter(c))
        {
            if (!targetCount.ContainsKey(c))
            {
                targetCount[c] = 0;
            }
            targetCount[c]++;
        }
    }

    string shortestWord = null;

    foreach (string word in words)
    {
        // Создаем словарь для подсчета количества каждой буквы в текущем слове
        Dictionary<char, int> wordCount = new Dictionary<char, int>();

        foreach (char c in word.ToLower())
        {
            if (!wordCount.ContainsKey(c))
            {
                wordCount[c] = 0;
            }
            wordCount[c]++;
        }

        // Проверяем, удовлетворяет ли текущее слово требованиям
        bool isCompletingWord = true;
        foreach (var kvp in targetCount)
        {
            if (!wordCount.ContainsKey(kvp.Key) || wordCount[kvp.Key] <
kvp.Value)
            {
                isCompletingWord = false;
                break;
            }
        }

        // Если текущее слово удовлетворяет требованиям и короче найденного
ранее
        if (isCompletingWord && (shortestWord == null || word.Length <
shortestWord.Length))
        {
            shortestWord = word;
        }
    }

    return shortestWord;
}

```