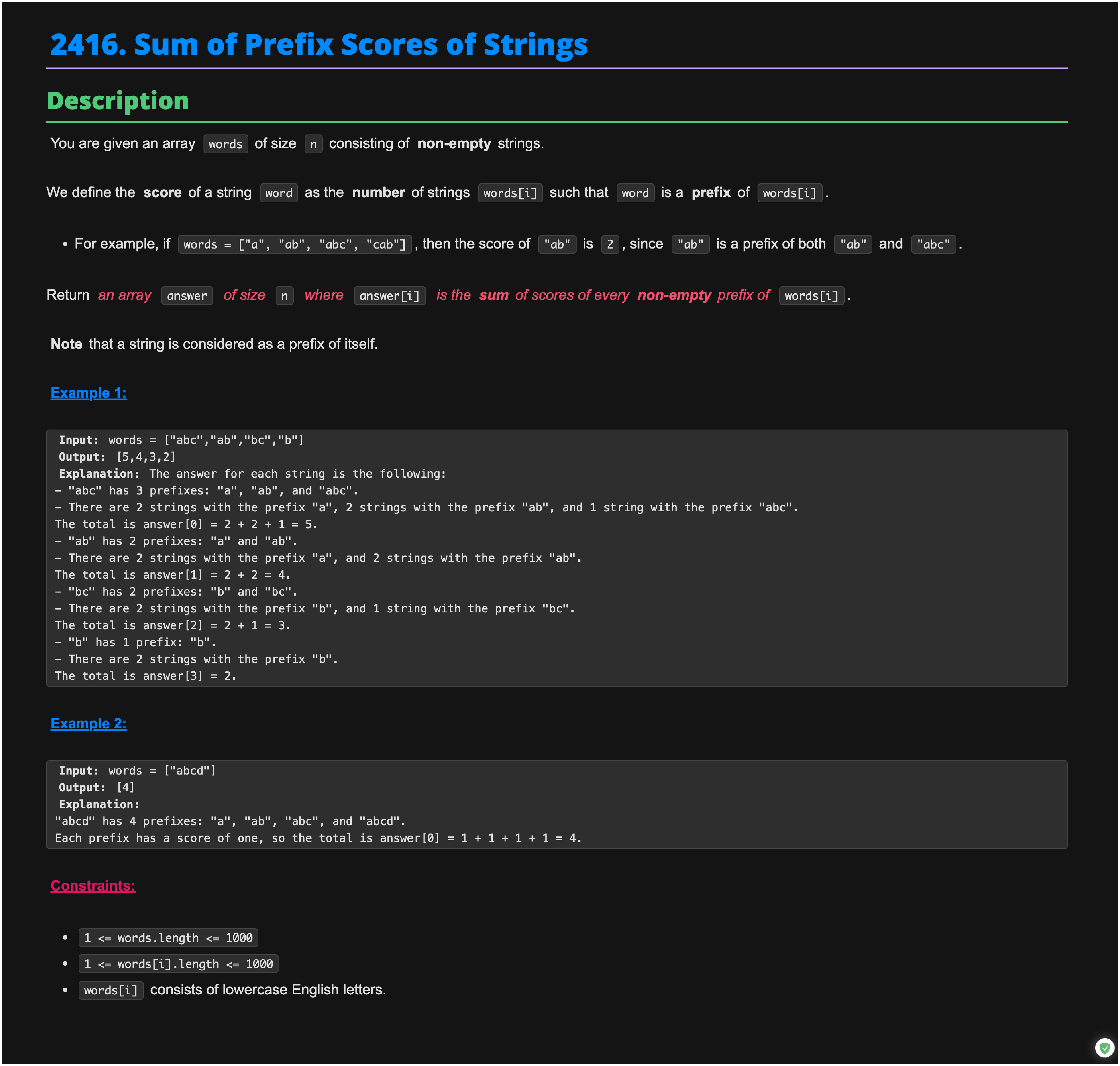

2416. Sum of Prefix Scores of Strings

Description

You are given an array `words` of size `n` consisting of **non-empty** strings.

We define the **score** of a string `word` as the **number** of strings `words[i]` such that `word` is a **prefix** of `words[i]`.

- For example, if `words = ["a", "ab", "abc", "cab"]`, then the score of `"ab"` is `2`, since `"ab"` is a prefix of both `"ab"` and `"abc"`.

Return *an array* `answer` *of size* `n` *where* `answer[i]` *is the* ***sum*** *of scores of every* ***non-empty*** *prefix of* `words[i]`.

Note that a string is considered as a prefix of itself.

Example 1:

```
 Input: words = ["abc","ab","bc","b"]
 Output: [5,4,3,2]
 Explanation: The answer for each string is the following:
- "abc" has 3 prefixes: "a", "ab", and "abc".
- There are 2 strings with the prefix "a", 2 strings with the prefix "ab", and 1 string with the prefix "abc".
The total is answer[0] = 2 + 2 + 1 = 5.
- "ab" has 2 prefixes: "a" and "ab".
- There are 2 strings with the prefix "a", and 2 strings with the prefix "ab".
The total is answer[1] = 2 + 2 = 4.
- "bc" has 2 prefixes: "b" and "bc".
- There are 2 strings with the prefix "b", and 1 string with the prefix "bc".
The total is answer[2] = 2 + 1 = 3.
- "b" has 1 prefix: "b".
- There are 2 strings with the prefix "b".
The total is answer[3] = 2.
```

Example 2:

```
 Input: words = ["abcd"]
 Output: [4]
 Explanation:
"abcd" has 4 prefixes: "a", "ab", "abc", and "abcd".
Each prefix has a score of one, so the total is answer[0] = 1 + 1 + 1 + 1 = 4.
```

Constraints:

- `1 <= words.length <= 1000`
- `1 <= words[i].length <= 1000`
- `words[i]` consists of lowercase English letters.