

Problem 3292: Minimum Number of Valid Strings to Form Target II

Problem Information

Difficulty: Hard

Acceptance Rate: 19.26%

Paid Only: No

Tags: Array, String, Binary Search, Dynamic Programming, Segment Tree, Rolling Hash, String Matching, Hash Function

Problem Description

You are given an array of strings `words` and a string `target`.

A string `x` is called **valid** if `x` is a prefix of **any** string in `words`.

Return the **minimum** number of **valid** strings that can be _concatenated_ to form `target`. If it is **not** possible to form `target`, return `-1`.

Example 1:

Input: words = ["abc", "aaaaa", "bcdef"], target = "aabcdabc"

Output: 3

Explanation:

The target string can be formed by concatenating:

* Prefix of length 2 of `words[1]`, i.e. `aa`. * Prefix of length 3 of `words[2]`, i.e. `bcd`. * Prefix of length 3 of `words[0]`, i.e. `abc`.

Example 2:

Input: words = ["abababab", "ab"], target = "ababaababa"

****Output:**** 2

****Explanation:****

The target string can be formed by concatenating:

* Prefix of length 5 of `words[0]` , i.e. ` "ababa"`. * Prefix of length 5 of `words[0]` , i.e. ` "ababa"`. *

****Example 3:****

****Input:**** words = ["abcdef"], target = "xyz"

****Output:**** -1

****Constraints:****

* `1 <= words.length <= 100` * `1 <= words[i].length <= 5` * `104` * The input is generated such that `sum(words[i].length) <= 105` . * `words[i]` consists only of lowercase English letters. * `1 <= target.length <= 5` * `target` consists only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
    int minValidStrings(vector<string>& words, string target) {

    }
};
```

Java:

```
class Solution {
public int minValidStrings(String[] words, String target) {

    }
}
```

Python3:

```
class Solution:  
    def minValidStrings(self, words: List[str], target: str) -> int:
```