Longest String Chain

You are given an array of words where each word consists of lowercase English letters.

word$_A$ is a **predecessor** of word$_B$ if and only if we can insert **exactly one** letter anywhere in word$_A$ **without changing the order of the other characters** to make it equal to word$_B$.

- For example, "abc" is a **predecessor** of "abac", while "cba" is not a **predecessor** of "bcad".

A **word chain** is a sequence of words [word$_1$, word$_2$, ..., word$_k$] with k >= 1, where word$_1$ is a **predecessor** of word$_2$, word$_2$ is a **predecessor** of word$_3$, and so on. A single word is trivially a **word chain** with k == 1.

Return *the **length** of the **longest possible word chain** with words chosen from the given list of* words.

Example 1:

Input: words = ["a","b","ba","bca","bda","bdca"]

Output: 4

Explanation: One of the longest word chains is ["a","ba","bda","bdca"].

Example 2:

Input: words = ["xbc","pcxbcf","xb","cxbc","pcxbc"]

Output: 5

Explanation: All the words can be put in a word chain ["xb", "xbc", "cxbc", "pcxbc", "pcxbcf"].

Example 3:

Input: words = ["abcd","dbqca"]

Output: 1

Explanation: The trivial word chain ["abcd"] is one of the longest word chains.

["abcd","dbqca"] is not a valid word chain because the ordering of the letters is changed.

Constraints:

- 1 <= words.length <= 1000
- 1 <= words[i].length <= 16
- words[i] only consists of lowercase English letters.