5542. Number of Ways to Form a Target String Given a Dictionary

My Submissions (/contest/biweekly-contest-38/problems/number-of-ways-to-form-a-target-string-given-a-dictionary/submissions/)

Back to Contest (/contest/biweekly-contest-38/)

You are given a list of strings of the **same length** `words` and a string `target`.

Your task is to form `target` using the given `words` under the following rules:

- `target` should be formed from left to right.
- To form the ith character (**0-indexed**) of `target`, you can choose the kth character of the jth string in `words` if `target[i] = words[j][k]`.
- Once you use the kth character of the jth string of `words`, you **can no longer** use the xth character of any string in `words` where `x <= k`. In other words, all characters to the left of or at index `k` become unusuable for every string.
- Repeat the process until you form the string `target`.

Notice that you can use **multiple characters** from the **same string** in `words` provided the conditions above are met.

Return *the number of ways to form* `target` *from* `words`. Since the answer may be too large, return it **modulo** $10^9 + 7$.

User Accepted:	14
User Tried:	26
Total Accepted:	14
Total Submissions:	32
Difficulty:	Hard

Example 1:

```
Input: words = ["acca","bbbb","caca"], target = "aba"
Output: 6
Explanation: There are 6 ways to form target.
"aba" -> index 0 ("acca"), index 1 ("bbbb"), index 3 ("caca")
"aba" -> index 0 ("acca"), index 2 ("bbbb"), index 3 ("caca")
"aba" -> index 0 ("acca"), index 1 ("bbbb"), index 3 ("acca")
"aba" -> index 0 ("acca"), index 2 ("bbbb"), index 3 ("acca")
"aba" -> index 1 ("caca"), index 2 ("bbbb"), index 3 ("acca")
"aba" -> index 1 ("caca"), index 2 ("bbbb"), index 3 ("caca")
```

Example 2:

```
Input: words = ["abba","baab"], target = "bab"
Output: 4
Explanation: There are 4 ways to form target.
"bab" -> index 0 ("baab"), index 1 ("baab"), index 2 ("abba")
"bab" -> index 0 ("baab"), index 1 ("baab"), index 3 ("baab")
"bab" -> index 0 ("baab"), index 2 ("baab"), index 3 ("baab")
"bab" -> index 1 ("abba"), index 2 ("baab"), index 3 ("baab")
```

Example 3:

```
Input: words = ["abcd"], target = "abcd"
Output: 1
```

Example 4:

```
Input: words = ["abab","baba","abba","baab"], target = "abba"
Output: 16
```

Constraints:

- `1 <= words.length <= 1000`
- `1 <= words[i].length <= 1000`
- All strings in `words` have the same length.
- `1 <= target.length <= 1000`
- `words[i]` and `target` contain only lowercase English letters.

Java

```
1  class Solution {
2      public int numWays(String[] words, String target) {
3
4      }
5  }
```

☐ **Custom Testcase** [Use Example Testcases]

▶ Run   ☁ Submit