Problem 820: Short Encoding of Words

Problem Information

Difficulty: Medium
Acceptance Rate: 60.72%
Paid Only: No
Tags: Array, Hash Table, String, Trie

Problem Description

A **valid encoding** of an array of `words` is any reference string `s` and array of indices `indices` such that:

* `words.length == indices.length` * The reference string `s` ends with the `'#'` character. * For each index `indices[i]`, the **substring** of `s` starting from `indices[i]` and up to (but not including) the next `'#'` character is equal to `words[i]`.

Given an array of `words`, return _the**length of the shortest reference string** _`s` _possible of any**valid encoding** of _`words` _._

Example 1:

Input: words = ["time", "me", "bell"] **Output:** 10 **Explanation:** A valid encoding would be s = "time#bell#" and indices = [0, 2, 5]. words[0] = "time", the substring of s starting from indices[0] = 0 to the next '#' is underlined in "_time_ #bell#" words[1] = "me", the substring of s starting from indices[1] = 2 to the next '#' is underlined in "ti _me_ #bell#" words[2] = "bell", the substring of s starting from indices[2] = 5 to the next '#' is underlined in "time#_bell_ #"

Example 2:

Input: words = ["t"] **Output:** 2 **Explanation:** A valid encoding would be s = "t#" and indices = [0].

Constraints:

* `1 <= words.length <= 2000` * `1 <= words[i].length <= 7` * `words[i]` consists of only lowercase letters.

Code Snippets

C++:

```
class Solution {
public:
int minimumLengthEncoding(vector<string>& words) {

}
};
```

Java:

```
class Solution {
public int minimumLengthEncoding(String[] words) {

}
}
```

Python3:

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