Problem 943: Find the Shortest Superstring

Problem Information

Difficulty: Hard
Acceptance Rate: 44.88%
Paid Only: No
Tags: Array, String, Dynamic Programming, Bit Manipulation, Bitmask

Problem Description

Given an array of strings `words`, return _the smallest string that contains each string in_ `words` _as a substring_. If there are multiple valid strings of the smallest length, return **any of them**.

You may assume that no string in `words` is a substring of another string in `words`.

Example 1:

Input: words = ["alex","loves","leetcode"] **Output:** "alexlovesleetcode" **Explanation:** All permutations of "alex","loves","leetcode" would also be accepted.

Example 2:

Input: words = ["catg","ctaagt","gcta","ttca","atgcatc"] **Output:** "gctaagttcatgcatc"

Constraints:

* `1 <= words.length <= 12` * `1 <= words[i].length <= 20` * `words[i]` consists of lowercase English letters. * All the strings of `words` are **unique**.

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public String shortestSuperstring(String[] words) {

}
}
```

Python3:

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