

Problem 1239: Maximum Length of a Concatenated String with Unique Characters

Problem Information

Difficulty: Medium

Acceptance Rate: 54.43%

Paid Only: No

Tags: Array, String, Backtracking, Bit Manipulation

Problem Description

You are given an array of strings `arr`. A string `s` is formed by the **concatenation** of a **subsequence** of `arr` that has **unique characters**.

Return _the**maximum** possible length_ of `s`.

A **subsequence** is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.

Example 1:

Input: arr = ["un", "iq", "ue"] **Output:** 4 **Explanation:** All the valid concatenations are: -
"" - "un" - "iq" - "ue" - "uniq" ("un" + "iq") - "ique" ("iq" + "ue") Maximum length is 4.

Example 2:

Input: arr = ["cha", "r", "act", "ers"] **Output:** 6 **Explanation:** Possible longest valid concatenations are "chaers" ("cha" + "ers") and "acters" ("act" + "ers").

Example 3:

Input: arr = ["abcdefghijklmnopqrstuvwxyz"] **Output:** 26 **Explanation:** The only string in arr has all 26 characters.

Constraints:

`* `1 <= arr.length <= 16` * `1 <= arr[i].length <= 26` * `arr[i]` contains only lowercase English letters.`

Code Snippets

C++:

```
class Solution {  
public:  
    int maxLength(vector<string>& arr) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int maxLength(List<String> arr) {  
  
    }  
}
```

Python3:

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