Problem 1980: Find Unique Binary String

Problem Information

Difficulty: Medium
Acceptance Rate: 79.42%
Paid Only: No
Tags: Array, Hash Table, String, Backtracking

Problem Description

Given an array of strings `nums` containing `n` **unique** binary strings each of length `n`, return _a binary string of length_`n` _that**does not appear** in _`nums` _. If there are multiple answers, you may return**any** of them_.

Example 1:

Input: nums = ["01","10"] **Output:** "11" **Explanation:** "11" does not appear in nums. "00" would also be correct.

Example 2:

Input: nums = ["00","01"] **Output:** "11" **Explanation:** "11" does not appear in nums. "10" would also be correct.

Example 3:

Input: nums = ["111","011","001"] **Output:** "101" **Explanation:** "101" does not appear in nums. "000", "010", "100", and "110" would also be correct.

Constraints:

* `n == nums.length` * `1 <= n <= 16` * `nums[i].length == n` * `nums[i] `is either `'0'` or `'1'`. * All the strings of `nums` are **unique**.

Code Snippets

C++:

```
class Solution {
public:
string findDifferentBinaryString(vector<string>& nums) {

}
};
```

Java:

```
class Solution {
public String findDifferentBinaryString(String[] nums) {

}
}
```

Python3:

```
class Solution:
def findDifferentBinaryString(self, nums: List[str]) -> str:
```