1980. Find Unique Binary String

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.
```

```python
class Solution:
    def findDifferentBinaryString(self, nums: List[str]) -> str:
        res = []
        n = len(nums[0])
        nums = set(nums)
        self.findDifferentBinaryStringUtil(nums,res,n,'')
        return res[0]
    def findDifferentBinaryStringUtil(self,nums,res,n,ssf):
        if len(ssf)==n:
            if ssf not in nums:
                res.append(ssf)
            return

        if len(res)==0:
            self.findDifferentBinaryStringUtil(nums,res,n,'0'+ssf)
            self.findDifferentBinaryStringUtil(nums,res,n,'1'+ssf)
        else:
            return
```