



@codestorywithmik

(Instagram, Facebook)

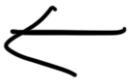
cswithMIK → Twitter

codestorywithMIK → whatsapp



STRINGS

Video-27



Leetcode
-1980 ✓✓

Medium ~~X~~

Find Unique

Binary Strings

Asked By :-

Meta

amazon

Google

1980. Find Unique Binary String

Medium

1.1K

45



Companies

Hint

$\Rightarrow \{ '010', '101', '000' \} \quad n=3$

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"]` $\rightarrow 2$

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.

$$1 \leq n \leq 16$$

Approach - 1

set

$\{ '01', '10' \} \quad n=2$

1, 2

¹⁴
↓
temp. substr (16-n, n) ; → "//"

Slight optimisation:-

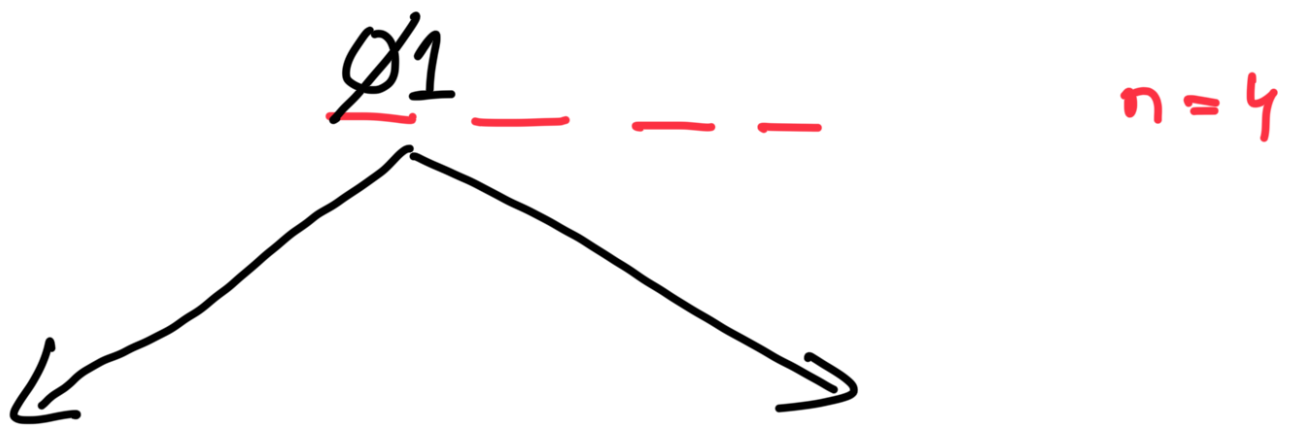
for (num = 0; num <= n ; num++)
{



Approach - 2 -

① Generate all strings
of length $\rightarrow n$

\therefore Recursion (Backtra(C)).



Approach-3.



nums = { "⁰1¹1¹", "¹0¹1¹", "²00¹1¹" } $n=3$

(Note: In the original image, arrows point to the first digit of each string: '1' in '111', '0' in '011', and '0' in '001'. The superscripts 0, 1, 2 are above the second and third digits respectively.)

Result = [0 0 0]



(n=3)

nums[0][0]

nums[i][1]

nums[2][2]

```
for(i = 0; i < n; i++) {
```

```
    char ch = nums[i][i];
```

```
    if (ch == '0')
```

```
        result += "1";
```

```
    else
```

```
        result += "0";
```

```
}
```

..

here

$O(n)$

S.C = $O(1)$

