Problem 3211: Generate Binary Strings Without Adjacent Zeros

Problem Information

Difficulty: Medium
Acceptance Rate: 87.89%
Paid Only: No
Tags: String, Backtracking, Bit Manipulation

Problem Description

You are given a positive integer `n`.

A binary string `x` is **valid** if all substrings of `x` of length 2 contain **at least** one `"1"`.

Return all **valid** strings with length `n`**,** in _any_ order.

Example 1:

Input: n = 3

Output: ["010","011","101","110","111"]

Explanation:

The valid strings of length 3 are: `"010"`, `"011"`, `"101"`, `"110"`, and `"111"`.

Example 2:

Input: n = 1

Output: ["0","1"]

Explanation:

The valid strings of length 1 are: `"0"` and `"1"`.

Constraints:

* `1 <= n <= 18`

Code Snippets

C++:

```
class Solution {
public:
vector<string> validStrings(int n) {

}
};
```

Java:

```
class Solution {
public List<String> validStrings(int n) {

}
}
```

Python3:

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