Problem 89: Gray Code

Problem Information

Difficulty: Medium
Acceptance Rate: 63.22%
Paid Only: No
Tags: Math, Backtracking, Bit Manipulation

Problem Description

An **n-bit gray code sequence** is a sequence of `2n` integers where:

* Every integer is in the **inclusive** range `[0, 2n - 1]`, * The first integer is `0`, * An integer appears **no more than once** in the sequence, * The binary representation of every pair of **adjacent** integers differs by **exactly one bit** , and * The binary representation of the **first** and **last** integers differs by **exactly one bit**.

Given an integer `n`, return _any valid**n-bit gray code sequence**_.

Example 1:

Input: n = 2 **Output:** [0,1,3,2] **Explanation:** The binary representation of [0,1,3,2] is [00,01,11,10]. - 0 _0_ and 0 _1_ differ by one bit - _0_ 1 and _1_ 1 differ by one bit - 1 _1_ and 1 _0_ differ by one bit - _1_ 0 and _0_ 0 differ by one bit [0,2,3,1] is also a valid gray code sequence, whose binary representation is [00,10,11,01]. - _0_ 0 and _1_ 0 differ by one bit - 1 _0_ and 1 _1_ differ by one bit - _1_ 1 and _0_ 1 differ by one bit - 0 _1_ and 0 _0_ differ by one bit

Example 2:

Input: n = 1 **Output:** [0,1]

Constraints:

* `1 <= n <= 16`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public List<Integer> grayCode(int n) {

}
}
```

Python3:

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