Counting Bits

Problem Statement

Given an integer **n**, return an array `ans` of size `n + 1` where:

```
ans[i] = number of 1's (set bits) in the binary representation of i
```

Example

```
Input: n = 5

Binary values:
0 → 0       → 0 ones
1 → 1       → 1 one
2 → 10      → 1 one
3 → 11      → 2 ones
4 → 100     → 1 one
5 → 101     → 2 ones

Output: [0,1,1,2,1,2]
```

Best Approach (Dynamic Programming – O(n))

Key Idea

For any number **i**:

```
i >> 1  → removes last bit
i & 1   → tells if last bit is 1 or 0
```

So formula becomes:

```
ans[i] = ans[i >> 1] + (i & 1)
```

Meaning:

- Count bits of half the number
- Add 1 if last bit is 1

Java Code

```java
class Solution {
    public int[] countBits(int n) {
        int[] ans = new int[n + 1];

        for (int i = 1; i <= n; i++) {
            ans[i] = ans[i >> 1] + (i & 1);
        }

        return ans;
    }
}
```