

# 0191. 位1的个数

👤 ITCharge ⌚ 大约 1 分钟

- 标签：位运算、分治
- 难度：简单

## 题目链接

- [0191. 位1的个数 - 力扣](#)

## 题目大意

**描述：**给定一个无符号整数  $n$ 。

**要求：**统计其对应二进制表达式中 1 的个数。

**说明：**

- 输入必须是长度为 32 的二进制串。

**示例：**

- 示例 1:

输入:  $n = 00000000000000000000000000001011$

输出: 3

解释: 输入的二进制串 `00000000000000000000000000001011` 中, 共有三位为 '1'。

py

- 示例 2:

输入:  $n = 000000000000000000000000000010000000$

输出: 1

解释: 输入的二进制串 `000000000000000000000000000010000000` 中, 共有一位为 '1'。

py

## 解题思路

### 思路 1：循环按位计算

1. 对整数  $n$  的每一位进行按位与运算，并统计结果。

### 思路 1：代码

```
class Solution:
    def hammingWeight(self, n: int) -> int:
        ans = 0
        while n:
            ans += (n & 1)
            n = n >> 1
        return ans
```

py

### 思路 1：复杂度分析

- 时间复杂度： $O(k)$ ，其中  $k$  是二进位的位数， $k = 32$ 。
- 空间复杂度： $O(1)$ 。

### 思路 2：改进位运算

利用  $n \& (n - 1)$ 。这个运算刚好可以将  $n$  的二进制中最低位的 1 变为 0。

比如  $n = 6$  时， $6 = 110_{(2)}$ ， $6 - 1 = 101_{(2)}$ ， $110 \& 101 = 100$ 。

利用这个位运算，不断的将  $n$  中最低位的 1 变为 0，直到  $n$  变为 0 即可，其变换次数就是我们要求的结果。

## 思路 2：代码

```
class Solution:
    def hammingWeight(self, n: int) -> int:
        ans = 0
        while n:
            n = n & (n - 1)
            ans += 1
        return ans
```

py

## 思路 2：复杂度分析

- 时间复杂度： $O(\log n)$ 。
- 空间复杂度： $O(1)$ 。