

0231.2 的幂

👤 [ITCharge](#) ⌚ 大约 1 分钟

- 标签：位运算、递归、数学
- 难度：简单

题目链接

- [0231.2 的幂 - 力扣](#)

题目大意

描述： 给定一个整数 n 。

要求： 判断该整数 n 是否是 2 的幂次方。如果是，返回 `True`；否则，返回 `False`。

说明：

- $-2^{31} \leq n \leq 2^{31} - 1$

示例：

- 示例 1:

输入: $n = 1$
输出: `True`
解释: $2^0 = 1$

py

- 示例 2:

输入: $n = 16$
输出: `True`
解释: $2^4 = 16$

py

解题思路

思路 1：循环判断

1. 不断判断 n 是否能整除 2。
 1. 如果不能整除，则返回 `False`。
 2. 如果能整除，则让 n 整除 2，直到 $n < 2$ 。
2. 如果最后 $n == 1$ ，则返回 `True`，否则则返回 `False`。

思路 1：代码

```
class Solution:
    def isPowerOfTwo(self, n: int) -> bool:
        if n <= 0:
            return False

        while n % 2 == 0:
            n //= 2
        return n == 1
```

py

思路 1：复杂度分析

- 时间复杂度： $O(\log_2 n)$ 。
- 空间复杂度： $O(1)$ 。

思路 2：数论判断

因为 n 能取的最大值为 $2^{31} - 1$ 。我们可以计算出：在 n 的范围内，2 的幂次方最大为 $2^{30} = 1073741824$ 。

因为 2 为质数，则 2^{30} 的除数只有 $2^0, 2^1, \dots, 2^{30}$ 。所以如果 n 为 2 的幂次方，则 2^{30} 肯定能被 n 整除，直接判断即可。

思路 2：代码

```
class Solution:
    def isPowerOfTwo(self, n: int) -> bool:
        return n > 0 and 1073741824 % n == 0
```

py

思路 2：复杂度分析

- 时间复杂度： $O(1)$ 。
- 空间复杂度： $O(1)$ 。