342 Power of Four

2018年4月9日 15:00

Question:

Given an integer (signed 32 bits), write a function to check whether it is a power of 4.

Example:

Given num = 16, return true. Given num = 5, return false.

Follow up: Could you solve it without loops/recursion?

来自 <https://leetcode.com/problems/power-of-four/description/>

给定一个整数 (32位有符整数型)，请写出一个函数来检验它是否是4的幂。

示例:

当 num = 16 时 ，返回 true 。 当 num = 5时，返回 false。

问题进阶：你能不使用循环/递归来解决这个问题吗？

Solution for Python3:

```python
class Solution1:
    def isPowerOfFour(self, num):
        """
        :type num: int
        :rtype: bool
        """
        import math
        return num > 0 and not math.log2(num) % 2

class Solution2:
    def isPowerOfFour(self, num):
        """
        :type num: int
        :rtype: bool
        """
        return num > 0 and (num & (num - 1)) == 0 and (num & 0x55555555) == num
```

Solution for C++:

```cpp
class Solution1 {
public:
    bool isPowerOfFour(int num) {
        return (num > 0) && int(log2(num)) == log2(num) && (int(log2(num)) % 2
== 0);
    }
};

class Solution2 {
public:
    bool isPowerOfFour(int num) {
        return num > 0 && (num & (num - 1)) == 0 && (num - 1) % 3 == 0;
    }
};

class Solution3 {
public:
    bool isPowerOfFour(int num) {
        return num > 0 && (num & (num - 1)) == 0 && (num & 0x55555555) == num;
    }
};
```

Appendix:

num & (num - 1) 去掉num二进制表示时最右边的一个1