762 Prime Number of Set Bits in Binary Representation

2018年5月4日 15:41

Given two integers L and R, find the count of numbers in the range [L, R] (inclusive) having a prime number of set bits in their binary representation.

(Recall that the number of set bits an integer has is the number of 1s present when written in binary. For example, 21 written in binary is 10101 which has 3 set bits. Also, 1 is not a prime.)

Example 1:

Input: L = 6, R = 10

Output: 4

Explanation:

6 -> 110 (2 set bits, 2 is prime)

7 -> 111 (3 set bits, 3 is prime)

9 -> 1001 (2 set bits , 2 is prime)

10->1010 (2 set bits , 2 is prime)

Example 2:

Input: L = 10, R = 15

Output: 5

Explanation:

10 -> 1010 (2 set bits, 2 is prime)

11 -> 1011 (3 set bits, 3 is prime)

12 -> 1100 (2 set bits, 2 is prime)

13 -> 1101 (3 set bits, 3 is prime)

14 -> 1110 (3 set bits, 3 is prime)

15 -> 1111 (4 set bits, 4 is not prime)

Note:

1. L, R will be integers L <= R in the range [1, 10^6].
2. R - L will be at most 10000.

来自 <https://leetcode.com/problems/prime-number-of-set-bits-in-binary-representation/description/>

给定两个整数 L 和 R ，找到闭区间 [L, R] 范围内，计算置位位数为质数的整数个数。

（注意，计算置位代表二进制表示中1的个数。例如 21 的二进制表示 10101 有 3 个计算置位。还有，1 不是质数。）

示例 1:

输入: L = 6, R = 10

输出: 4

解释:

6 -> 110 (2 个计算置位，2 是质数)

7 -> 111 (3 个计算置位，3 是质数)

9 -> 1001 (2 个计算置位，2 是质数)

10-> 1010 (2 个计算置位，2 是质数)

示例 2:

输入: L = 10, R = 15

输出: 5

解释:

10 -> 1010 (2 个计算置位, 2 是质数)

11 -> 1011 (3 个计算置位, 3 是质数)

12 -> 1100 (2 个计算置位, 2 是质数)

13 -> 1101 (3 个计算置位, 3 是质数)

14 -> 1110 (3 个计算置位, 3 是质数)

15 -> 1111 (4 个计算置位, 4 不是质数)

注意:

1. L, R 是 L <= R 且在 [1, 10^6] 中的整数。
2. R - L 的最大值为 10000。

Solution for Python3:

```python
class Solution1:
    def countPrimeSetBits(self, L, R):
        """
        :type L: int
        :type R: int
        :rtype: int
        """
        prime = [2,3,5,7,11,13,17,19]
        ans = 0
        for n in range(L, R + 1):
            if bin(n).count('1') in prime:
                ans += 1
        return ans

class Solution2:
    def countPrimeSetBits(self, L, R):
        """
        :type L: int
        :type R: int
        :rtype: int
        """
        primes = [2,3,5,7,11,13,17,19]
        return sum(bin(n).count('1') in primes for n in range(L, R+1))

# 665772的二进制表示：10100010100010101100
# 只有右移质数个位后最右位值恰好是1
#int('10100010100010101100', 2)=665772
#2^19<10^6<2^20
class Solution3:
    def countPrimeSetBits(self, L, R):
        """
        :type L: int
        :type R: int
        :rtype: int
        """
        return sum(665772 >> bin(i).count('1') & 1 for i in range(L, R+1))
```

Solution for C++:

```cpp
class Solution1 {
public:
    int countPrimeSetBits(int L, int R) {
        unordered_set<int> s{2,3,5,7,11,13,17,19};
        int ans = 0;
        for (int i = L; i <= R; i++) {
            if (s.count(bitset<32>(i).count()))
                ans++;
        }
        return ans;
    }
};
```

```cpp

class Solution2 {
public:
    int countPrimeSetBits(int L, int R) {
        int ans = 0;
        while (L <= R) {
             ans += 665772 >> bitset<32>(L++).count() & 1;
        }
        return ans;
    }
};
```