191 Number of 1 Bits

2018年4月4日 14:40

Question:

Write a function that takes an unsigned integer and returns the number of ’1' bits it has (also known as the Hamming weight).
For example, the 32-bit integer ’11' has binary representation 00000000000000000000000000001011, so the function should return 3.

来自 <https://leetcode.com/problems/number-of-1-bits/description/>

编写一个函数，输入是一个无符号整数，返回的是它所有 位1 的个数（也被称为汉明重量）。
例如，32位整数 ’11’ 的二进制表示为 00000000000000000000000000001011，所以函数返回3。

Solution for Python3:

```python
class Solution1(object):
    def hammingWeight(self, n):
        """
        :type n: int
        :rtype: int
        """
        cnt = 0;
        while n:
           cnt += n & 1
           n >>= 1
        return cnt

class Solution2(object):
    def hammingWeight(self, n):
        """
        :type n: int
        :rtype: int
        """
        cnt = 0;
        while n:
           cnt += 1
           n &= n - 1
        return cnt
```

Solution for C++:

```cpp
class Solution1 {
public:
    int hammingWeight(uint32_t n) {
        int cnt = 0;
        while (n) {
            cnt += n & 1;
            n >>= 1;
        }
        return cnt;
    }
};
```

```cpp

class Solution2 {
public:
    int hammingWeight(uint32_t n) {
        int cnt = 0;
        while (n) {
            cnt++;
            n &= n - 1;
        }
        return cnt;
    }
};
```

Appendix:

n &= n - 1: 每次会移除n最右边的一个1

n >>= 1: 每次会移除n最右边的一位