

191 位1的个数

Label: 位运算

编写一个函数，输入是一个无符号整数（以二进制串的形式），返回其二进制表达式中数字位数为 '1' 的个数（也被称为汉明重量）。

输入必须是长度为 32 的二进制串

- 迭代

```
public class Solution {
    // you need to treat n as an unsigned value
    public int hammingWeight(int n) {
        int count = 0;
        for (char c : Integer.toBinaryString(n).toCharArray()) {
            if (c == '1') {
                count++;
            }
        }
        return count;
    }
}
```

- 位运算 每次判断最低位是否为1 算数位移，弃最低位，高位用符号位填补

```
public class Solution {
    // you need to treat n as an unsigned value
    public int hammingWeight(int n) {
        int count = 0;
        for (int i = 0; i < 32; ++i) { // 不使用这个 while (n != 0)
            count += n & 1;
            n >>= 1; // 右移一位
        }
        return count;
    }
}
```

- 位运算 每次判断最低位是否为1 逻辑位移，舍弃最低位，高位用 0 填补

```
public class Solution {
    // you need to treat n as an unsigned value
    public int hammingWeight(int n) {
        int count = 0;
        while (n != 0) {
            count += n & 1;
            n >>= 1;
        }
        return count;
    }
}
```

- 位运算 $n \& n-1$

```
public class Solution {  
    // you need to treat n as an unsigned value  
    public int hammingWeight(int n) {  
        int res = 0;  
        while (n != 0) {  
            n &= n - 1; // 每次都会消除n中最后一个1，那么消除几次就说明原始n有几个1  
            res += 1;  
        }  
        return res;  
    }  
}
```