# 题目

两个整数之间的汉明距离指的是这两个数字对应二进制位不同的位置的数目。

给出两个整数 x 和 y，计算它们之间的汉明距离。

**注意：**

0 ≤ x, y < 231.

**示例:**

**输入:** x = 1, y = 4

**输出:** 2

**解释:**

1 (0 0 0 1)

4 (0 1 0 0)

↑ ↑

上面的箭头指出了对应二进制位不同的位置。

# 分析

## 方法一：位操作

1、两个数字异或之后可以得到不同二进制位的数字（异或求不同）

2、计算该数字中1的个数，即是汉明距离

计算1的个数时，有几种方法:

1、不断和左移的1进行与，判断该位是否为1；

2、n&(n-1)就是把n最右边的bit 1位去掉，看能去掉几次，就有几个1位。

**代码：**

/\*

1.先经过异或操作求得两个数不同的数

2.再判断x^y结果中含有1的个数

\*/

class Solution {

public:

int hammingDistance(int x, int y) {

int tmp = x ^ y; //通过异或操作求出不同值

int num = 0;

for(int i=0;i<32;i++)

{

if(tmp & (1<<i))

num++;

}

return num;

}

};

## 方法二：n&(n-1)巧运算

思路：

代码：

class Solution {

public:

int hammingDistance(int x, int y) {

return countBitNumber(x^y);

}

int countBitNumber(int n)

{

int count = 0;

while(n)

{

n = (n & n-1);

++count;

}

return count;

}

};