461 Hamming Distance

2018年4月12日 11:15

The Hamming distance between two integers is the number of positions at which the corresponding bits are different.

Given two integers x and y, calculate the Hamming distance.

Note:

0 ≤ x, y < 2^{31}.

Example:

Input: x = 1, y = 4

Output: 2

Explanation:

1   (0 0 0 1)

4   (0 1 0 0)

       ↑   ↑

The above arrows point to positions where the corresponding bits are different.

来自 <https://leetcode.com/problems/hamming-distance/description/>

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 < 2^{31}.

Solution for Python3:

```
class Solution1:
    def hammingDistance(self, x, y):
        """
        :type x: int
        :type y: int
        :rtype: int
        """
        x = x ^ y
        y = 0
        while t:
```

```python
            y += 1
            x &= (x - 1)
        return y

class Solution2:
    def hammingDistance(self, x, y):
        """
        :type x: int
        :type y: int
        :rtype: int
        """
        return bin(x^y).count('1')
```

Solution for C++:

```cpp
class Solution1 {
public:
    int hammingDistance(int x, int y) {
        int res = 0;

            res += ((x % 2) != (y % 2));
            x >>= 1;
            y >>= 1;
        }
        return res;
    }
};

class Solution2 {
public:
    int hammingDistance(int x, int y) {
        int res = 0, n = x ^ y;
        while (n) {
            res++;
            n &= n - 1;
        }
        return res;
    }
};
```