

常用的几个异或操作

1. 利用或操作 `|` 和空格将英文字符转小写:

```
('a' | ' ') = 'a'
('A' | ' ') = 'a'
```

2. 利用与操作 `&` 和下划线将英文字符转大写:

```
('B' & '_') = 'B'
('b' & '_') = 'B'
```

3. 利用异或操作 `^` 和空格进行英文字符大小写互换:

```
('d' ^ ' ') = 'D'
('D' ^ ' ') = 'd'
```

4. 判断两个数是否异号:

```
int x = -1, y = 2;
bool f = ((x ^ y) < 0); // true

int x = 3, y = 4;
bool f = ((x ^ y) < 0); // false
```

5. 不用临时变量交换两个数:

```
int a = 1, b = 2;
a ^= b;
b ^= a;
a ^= b;
```

6. 加一

```
int n = 1;
n = -~n; // n = 2
```

7. 减一

```
int n = 2;
n = -~n; // n = 1
```

算法常用操作

$n \& (n-1)$ 常用来消除 n 的二进制位数中的最后一个1.

```
class Solution {
public:
    int hammingWeight(uint32_t n) {
        int res = 0;
        while (n != 0) {
            n = n & (n - 1);
            res++;
        }

        return res;
    }
};
```

判断一个数是不是2的指数

一个数如果是2的指数，那么他的二进制位一定只有一个1.

```
bool isPowerOfTwo(int n) {
    if(n <= 0) {
        return false;
    }

    return n & (n-1) == 0;
}
```

查找数组中仅出现一次的数

```
class Solution {
public:
    int singleNumber(std::vector<int>& nums) {
        int res = 0;
        for (auto item : nums) {
            res ^= item;
        }

        return res;
    }
};
```

