查阅更多的题解，请点击

Problem

136. Single Number

Given a non-empty array of integers, every element appears twice except for one. Find that single one.

Note:
Your algorithm should have a linear runtime complexity. Could you implement it without using extra memory?

Example 1:

```
Input: [2,2,1]
Output: 1
```

Example 2:

```
Input: [4,1,2,1,2]
Output: 4
```

Solution

由于题目要求在线性时间内完成，显然该题需要得出结果，必须知道全局的信息，即至少需要遍历一次数组。
比较直观的做法是在遍历的过程中保存已遍历的局部信息，新的数与局部信息进行比较：可以用hash表实现，数值作为hash表的key，遍历到下一个数时，若该key已存在，则删除该记录，继续遍历；若key不存在，则插入，继续遍历。最后遍历结束hash表中存在的key，即为题目所求解

实际上题目希望不使用额外的储存空间，则需要其他的解法：注意题目中所有的数仅有两种情况：成对出现或者是单独出现（仅一个数），而对于两个相同的数，按位异或，结果为0，则可以对数组中所有的数进行异或操作，最后的结果为所求解
GitHub传送门

XOR

```
class Solution
{
  public:
    int singleNumber(vector<int> &nums)
    {
        return std::accumulate(begin(nums), end(nums), 0, std::bit_xor<int>());
    }
};
```