1. Single Number III

Given an array of numbers nums, in which exactly two elements appear only once and all the other elements appear exactly twice. Find the two elements that appear only once.

**Example:**

Input: [1,2,1,3,2,5]  
Output: [3,5]

**Note**:

1. The order of the result is not important. So in the above example, [5, 3] is also correct.
2. Your algorithm should run in linear runtime complexity. Could you implement it using only constant space complexity?

**解** 在只有一个出现一次，其余数字出现两次的数组里面找出现一次的数字通过位异或即可实现。如果能将本题中的两个数字区分开来，问题也就可以解决了。思路同样是异或

* step1：生成mask，将所有的数字都进行异或，最后的结果就是两个出现了一次的数字（记为a, b）的异或结果，这两个数字至少有1位不相等，将异或结果的出现为1的位置作为mask，mask的二进制只有1位是1
* step2：寻找a和b，对于数组中的数字x，如果x&mask是1，说明x可能是a（或b）；如果x&mask是0，说明x可能是b（或a）

class Solution {  
public:  
 vector<int> singleNumber(vector<int>& nums) {  
 int x\_or = 0;  
 for(auto x: nums)x\_or ^= x;  
 int mask = 1;  
 while((mask & x\_or) == 0)mask <<= 1;  
 int ans1 = 0, ans2 = 0;  
 for(auto x : nums){  
 if((mask & x) != 0){  
 ans1 ^= x;  
 }else{  
 ans2 ^= x;  
 }  
 }  
 return {ans1, ans2};  
 }  
};