

136 只出现一次的数字

Label: 位运算 Hash

给定一个非空整数数组，除了某个元素只出现一次以外，其余每个元素均出现两次。找出那个只出现了一次的元素。

你的算法应该具有线性时间复杂度。 你可以不使用额外空间来实现吗？

输入: [4,1,2,1,2]

输出: 4

- 排序（进行奇偶对比） 排序之后用栈思路也比较清晰

```
class Solution {
    public int singleNumber(int[] nums) {

        if (nums.length == 1) {
            return nums[0];
        }

        Arrays.sort(nums);
        if (nums[0] != nums[1]) {
            return nums[0];
        } else if (nums[nums.length-1] != nums[nums.length-2]){
            return nums[nums.length - 1];
        }

        int mark = 0;
        for (int i = 0; i < nums.length - 1; i++) {
            mark = i ;
            if (nums[i] != nums[++i]) {
                break;
            }
        }

        return nums[mark];
    }
}
```

- 位运算

```
class Solution {
    public int singleNumber(int[] nums) {

        if (nums.length == 1) {
            return nums[0];
        }

        int single = 0; // 0与任何数异或都为任何数
        for (int i : nums) {
            single ^= i;
        }
        return single;
    }
}
```

- map

```
class Solution {
    public int singleNumber(int[] nums) {
        if (nums.length == 1) {
            return nums[0];
        }

        HashMap<Integer,Integer> map = new HashMap<>();
        int v = 0;
        for (int i = 0; i < nums.length; i++) {
            v = map.getOrDefault(nums[i],0);
            map.put(nums[i], ++v );
        }

        for (Integer key : map.keySet()) {
            if (map.get(key) == 1)
                return key;
        }
        return 0;
    }
}
```