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.

For example:

Given nums = [1, 2, 1, 3, 2, 5], return [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?

带测试提交版本

**import** java.util.\*;

**import** java.lang.\*;

**public** **class** SingleNumberThree {

**public** **int**[] singleNumber(**int**[] nums) {

HashSet<Integer> h = **new** HashSet();

**for**( **int** i= 0;i<nums.length;i++) {

**if**(!h.contains(nums[i])) {

h.add(nums[i]);

}

**else** h.remove(nums[i]);

}

Iterator i = h.iterator();

**int** m = h.size();

**int**[] result = **new** **int**[m] ;

**for** (**int** j=0;j<m;j++) {

result[j] = (**int**) i.next();

}

**return** result;

}

**public** **static** **void** main (String args[] ) {

**int** nums[]= {1,1,2,2,3,3,4,5,6,7,8,9};

SingleNumberThree sn3 = **new** SingleNumberThree();

**int** num[]= sn3.singleNumber(nums);

System.***out***.print(Arrays.*toString*(num));

}

}

