1、一个数组中除了一个元素只出现一次外，剩余元素均出现了2次，找到这个出现一次的元素。要求：在O(n)的时间O(1)的空间。

**思路：异或**

2、长度为n的数组循环左移k位。要求：O(n)的时间O(1)的空间。

**思路：[AT BT]T=[B A]**

3、蓄水池抽样。

4、一个数组，大小为n里面存储的数的大小为0-n-1，有些数出现多次，要求找出至少一个出现多次的数。要求：O(n)的时间，O(1)的空间。

**思路：尝试将所有的元素k都放到其对应的地方(索引为k处 ,使得v[k]=k)**

1 while(i<=n)

2 if v[i]≠i

3 if v[v[i]]==v[i]

4 value=v[i] and break**//该数出现了两次**

5 else

6 exchange v[v[i]] and v[i]**//会继续当前i的循环**

7 else

8 i++

5、子集和问题，从集合S中找出所有和为sum的子集（类似于动态规划的问题，**区别在于：当分割成一个数和另一个和的时候，该和可能是不可分的，也就是有些情况是不存在的，需要加以区分**）

1. 对集合S进行排序

2. 找出集合中小于sum的最大元素的索引n

3. 在索引1-n中（集合S的子集Sub）中寻找和为sum的子集

List<Set> L

SubSum(S,sum,L)

1 sum<S[1]

2 return

3 int n=MaxBeneathSum(S)

4 if sum==S[n]

5 L.add(sum,sum)

4 for i=1 to n

5 SubSum(S,sum-S[i],List)