[数组中找特定元素相关](http://blog.csdn.net/ohmygirl/article/details/7850518)

本次总结在数组中查找特定的元素。

碰到的面试题目主要包括：

**1.假如你有一个用1001个整数组成的数组，这些整数是任意排列的，但是你知道所有的整数都在1到1000之间（包括1000）、此外，除了一个数字出现两次外，其他的数字只出现了一次。假设你对数组做一次处理，用一种算法找出重复的那个数字，要求不使用大量额外的存储空间（即要求辅助空间为O(1)）**

    题目的意思很明显，1001个数字中只有一个数字出现了两次，其余的都是一次。并且不缺少任何数字。那么找到这个数字，我们可以有很多思路：

    思路a:求和，最简单也最容易想出来的方法 1+2+3+4+。。。+1000的和与数组的和求差，差的结果就是这个数字，缺点是，求和的话，结果可能会溢出。不妨看思路2

    思路b:异或，位操作总有让你惊喜的地方，原理是：

        @ 如果两个相同的数求异或，那么结果为0。

        @ 0与一个数异或的结果为这个数

        @ a^b ^a = b   ;a^b^b  = a;

       据此，可以对数组的元素与1,2,34,56,7,8...1000依次异或，最后的结果就是出现两次的数字。

相应的代码如下：

**[cpp]** [view plaincopy](http://blog.csdn.net/ohmygirl/article/details/7850518)

1. #include <stdio.h>
2. #include <stdlib.h>
4. **int** findTheNum(**int** \*a){
5. **int** k = a[0];
6. **for**(**int** i=1;i<=1000;i++){
7. k ^= (a[i]^i);
8. }
9. **return** k;
10. }
12. main(){
13. **int** a[1001];
14. **for**(**int** i = 0;i<=999;i++ ){
15. a[i] = i+1;
16. }
17. a[1000] = rand()%1000;
18. printf("the num : %d %d \n",a[1000],findTheNum(a));
19. **return** 0;
20. }

**2.腾讯的一道面试题：在一堆数字中，有两个数字出现了奇数次，其余的数字出现了偶数次，设计一种算法找出这两个数字。要求时间复杂度为O(n)，空间复杂度为O(1)**

       @[onlyonename](http://my.csdn.net/onlyonename" \t "_blank)

       假设这两个数为a，b，将数组中所有元素异或结果x=a^b，判断x中位为1的位数（注：因为a！=b，所以x！=0，我们只需知道某一个位为1的位数k，例如0010 1100，我们可取k=2或者3，或者5），然后将x与数组中第k位为1的数进行异或，异或结果就是a，b中一个，然后用x异或，就可以求出另外一个。

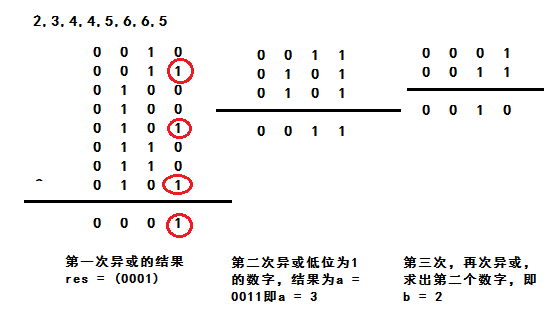
据此，不难写出代码：

**[cpp]** [view plaincopy](http://blog.csdn.net/ohmygirl/article/details/7850518)

1. #include <stdio.h>
2. **void** getNum(**int** a[],**int** len,**int** &one,**int** &two){
3. **int** res = 0;
4. **for**(**int** i = 0;i<len;i++){
5. res = res^a[i];
6. }
7. **int** temp = res;
8. **int** k = 0;
9. **while**(!(temp&1)){
10. temp = temp>>1;
11. k++;
12. }
13. temp = 0;
14. **for**(**int** i = 0;i < len;i++){
15. **if**((a[i]>>k)&1){
16. temp = temp^a[i];
17. }
18. }
19. one = temp ;
20. two = temp^res;
21. }
23. **int** main(){
24. **int** a[] = {2,3,4,4,5,6,6,5};
25. **int** one ,two;
26. getNum(a,8,one,two);
27. printf("%d --- %d \n",one,two);
28. **return** 0;
29. }

这题可以充分看到位运算的强大。

以该题为例，算法的扫描过程如下图所示：



**3。数组中有多个数字只出现了一次，且这些数字都在1000之间，用一种算法找出第一次只出现一次的数字，要求线性时间。（类似的有，在一个字符串中找到第一个只出现一次的字符）**

类似这种找“第一次出现一次的数字（字符）”的题目有一个共同的特点：数字或字符的范围是确定的。因此我们可以考虑计数。具体思路是：首先扫描一次原数组，记录每个数字的出现次数，然后再一次扫描原数组，并查询其出现的次数，一旦遇到出现次数为1的数字，那么就返回这个数字即可。相应的代码如下：

**[cpp]** [view plaincopy](http://blog.csdn.net/ohmygirl/article/details/7850518)

1. #include <stdio.h>
2. #include <string.h>
3. #define N 1000
5. **int** getFirstOne(**int** \*a,**int** n){
6. **int** b[N+1];
7. memset(b,0,N+1);
8. **for**(**int** i = 0;i < n;i++){
9. b[a[i]]++;
10. }
12. **for**(**int** i = 0;i < n;i++){
13. **if**(b[a[i]] == 1){
14. **return** a[i];
15. }
16. }
18. }
20. **int** main(){
21. **int** a[] = {1,2,1,2,3,4,5,4,5};
22. printf("%d \n",getFirstOne(a,9));
23. }

**4.一个输入流，很大*,*大到没有存储器可以将其存储下来，从这个输入流中随机取出m个元素，要求各个元素被取出的概率相同。输入流只能扫描一次。**

这个题目有一种算法叫做蓄水池抽样，思路大致是这样的，先选中前m个， 从第m+1个元素到最后一个元素为止， 以m/i  (i=m+1, m+2,...,N) 的概率选中第i个元素， 并且随机替换掉一个原先选中的元素， 这样遍历一次得到m个元素， 可以保证完全随机选取。

相应的代码如下：（这里只说明原理，注意rand()函数并不是真正意义上的随机数,具体可见：http://wenku.baidu.com/view/9bee261eb7360b4c2e3f6418.html）

**[cpp]** [view plaincopy](http://blog.csdn.net/ohmygirl/article/details/7850518)

1. #include <stdio.h>
2. #include <stdlib.h>
3. #define N 10000
4. #define M 20
6. **int** getRandM(**int** \*a,**int** n,**int** \*result,**int** m){
7. **int** i,j;
8. **for**(i = 0;i < m;i++){
9. result[i] = a[i];
10. }
12. **for**(j = i; j < n; j++){
13. **int** randN = rand()%j;
14. **if**(randN >= 0 && randN  < m){
15. **int** randChange = rand()%m;
16. result[randChange] = a[j];
17. }
18. }
19. }
21. **int** main(){
22. **int** inStream[N];
23. **for**( **int** i = 0;i < N;i++){
24. inStream[i] = i;
25. }
26. **int** outStream[M];
27. getRandM(inStream,N,outStream,M);
28. **for**(**int** i = 0;i<M;i++){
29. printf("%d ",outStream[i]);
30. }
31. printf("\n");
32. }

另：在有序数组中查找符合条件的两个数字的解法见：双指针法的应用： http://blog.csdn.net/ohmygirl/article/details/7850068

在数组中找k个数的和为m的算法。

其他有关数组中选取特定元素的题目待补充。