Two sum

Given an array of integers, return indices（目录index的复数） of the two numbers such that they add up to a specific target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

Example:

Given nums=[2,7,11,15],target=9,

Because nums[0]+nums[1]=2+7=9,return [0,1].

该题的时间复杂度要求为n。“时间复杂度不够用空间复杂度来换，anyway，trade off”

给定一个整数数组，返回两个数相加为一个特定的数字的这两个数的索引。你可以假设每一个输入只对应一个答案，且一个元素不能被使用两次。

类似题目：

[4Sum](http://www.cnblogs.com/grandyang/p/4515925.html)

[3Sum Smaller](http://www.cnblogs.com/grandyang/p/5235086.html)

[3Sum Closest](http://www.cnblogs.com/grandyang/p/4510984.html)

[3Sum](http://www.cnblogs.com/grandyang/p/4481576.html)

[Two Sum III - Data structure design](http://www.cnblogs.com/grandyang/p/5184143.html)

[Two Sum II - Input array is sorted](http://www.cnblogs.com/grandyang/p/5185815.html)

解法1：（1）先排个序？然后再判断最大的两个相加的值和目标值的大小，若相加后的值小于目标值，说明整个数组都没有一个值对满足这个条件，否则就开始判断哪两个数相加是等于目标值的。（2）接下来的步骤有序和无序的数组都可以使用，就是第一个数和后面的所有数相加看是否等于目标值，然后循环这一步骤。

public class two\_sum\_1{

public static void main(String[] args){

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

int var = 10;

int res1=0,res2=0,i=0,j=0;

labe:for(i = 0;i <= arr.length;i++){

for(j = 1;j<= arr.length;j++){

if(arr[i]+arr[j] == var){

res1 = i;

res2 = j;

break labe;

}

}

}

System.out.println("["+res1+","+res2+"]");

System.out.println("["+i+","+j+"]");

}

}

输出结果为：[0,8]

[0,8]

解法2：这个方法可以把排好序的数组的solution全部找出来，但是一旦一个数组没有排序，意味着break之后i+1，会漏掉一些solutions。

public class two\_sum\_1{

public static void main(String[] args){

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

int var = 10;

int i=0,j=0;

for(i = 0;i <= arr.length;i++){

for(j = 1;j<= arr.length;j++){

if(arr[i]+arr[j] == var){

System.out.println("["+i+","+j+"]");

break;

}

}

}

}

}

报错Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 9

at two\_sum\_1.main(two\_sum\_1.java:30)

解法3：用hashmap使得时间复杂度下降，空间复杂度置换空间复杂度。

class Solution{

public int[] twoSum(int[] nums,int target){

HashMap<Integer,Integer> m = new HashMap<Integer,Integer>;

int[] res = new int[2];

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

m.put(nums[i],i);

}

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

int t = target-nums[i];

if(m.containsKey(t)&&m.get(t) !=i){

res[0] = i;

res[1] =m.get(t);

break;

}

}

return res;

}

}

错误：

1. Java的类名起错：一开始起的是1\_two\_sum，后来改为two\_sum\_1，java类名。表示符由大小写字母，下划线，数字，$符号组成。开头可以是大小写字母，下划线，和$符号。数字是不能作为开头的。
2. Java变量命名规则：（1）变量：codeName，intcount，也就是说单个单词全部小写，多个单词组成的变量除了第一个单词小写，其余所有单词的首字母都要大写。
3. Java类名的命名规则：所有单词的首字母大写。
4. Public static void main(String[] args){}java的main函数忘记了。
5. 注意：java每一个{}括号内定义的变量的可用范围都只是在这个括号之内，括号之外就不行了。
6. 如何跳出多层循环：在多层循环外加一个表示符，然后break 该标识符名称。
7. 忘记如何使用System.out.println输出自己想要的格式。
8. HashMap用法：

HashMap<String,String> hashMap = new HashMap<String,String>();

hashMap.put("cn","中国");

hashMap.put("jp","日本");

hashMap.put("cn","中国");

System.out.println("cn:"+hashMap.get("cn"));//返回的是key=cn时value的值

System.out.println(hashMap.containsKey("cn"));//看hashmap里是否有cn，有返回true