1. 二维数组中的查找

在一个二维数组中，每一行都按照从左到右递增的顺序排序，每一列都按照从上到下递增的顺序排序。请完成一个函数，输入这样的一个二维数组和一个整数，判断数组中是否含有该整数。

public class Solution {

public boolean Find(int [][] array,int target) {

int columns=array.length-1;

int rows=0;

//以右上角为参照点，依次比较移动行列，缩小范围，直至找到对应的数或者循环完毕

while(columns>=0&&rows<array[0].length){

if(target<array[rows][columns]){

columns--; //左移

}else if(target>array[rows][columns]){

rows++; //下移

}else{

return true;

}

}

return false;

}

}

2.请实现一个函数，将一个字符串中的空格替换成“%20”。例如，当字符串为We Are Happy.则经过替换之后的字符串为We%20Are%20Happy。

法1：用java Sting API实现

public class Solution {

public String replaceSpace(StringBuffer str) {

String s=new String(str);

return s.replaceAll(" ","%20");

}

}

法2：自己写

class Solution {

public string replaceSpace(string str) {

if(str.size() == 0||str==null)

return str;

int orgStrLen = str.size();

int blankNum = 0;

int index = 0;

while(str[index] != '\0')

{

if(str[index] == ' ')

++blankNum;

++index;

}

if(blankNum == 0)

return str;

int newStrLen = orgStrLen + 2\*blankNum;

int pNew = newStrLen - 1;

int pOld = orgStrLen - 1;

str.resize(newStrLen, '0');

while(pOld >= 0 && pNew>pOld)

{

if(str[pOld] == ' ')

{

str[pNew--] = '0';

str[pNew--] = '2';

str[pNew--] = '%';

}

else

{

str[pNew--] = str[pOld];

}

--pOld;

}

return str;

}

};

3.输入一个链表，从尾到头打印链表每个节点的值。

/\*\*

\* public class ListNode {

\* int val;

\* ListNode next = null;

\*

\* ListNode(int val) {

\* this.val = val;

\* }

\* }

\*

\*/

import java.util.Stack;

import java.util.ArrayList;

public class Solution {

public ArrayList<Integer> printListFromTailToHead(ListNode listNode) {

ArrayList<Integer> list = new ArrayList<Integer>();

if(listNode == null){

return list;

}

Stack<Integer> stk = new Stack<Integer>();

while(listNode != null){

stk.push(listNode.val);

listNode = listNode.next;

}

while(!stk.isEmpty()){

list.add(stk.pop());

}

return list;

}

}