**《算法设计与分析》周练记录表-第1周**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **班学号** | **姓名** | **OJ账号** | **练习平台** | **完成题目数量** | **题号** |
| 11117233 | 徐鸿飞 | X\_cool | LeetCode | 5 | 27，150，20，283，19 |
|  | LintCode |  |  |
| HOXU | 牛客网 | 3 | 1，2，15 |
|  | CCF-CSP |  |  |

LeetCode：https://leetcode.com/ 或https://leetcode-cn.com/

LintCode：https://www.lintcode.com/

牛客网（剑指Offer）：https://www.nowcoder.com/ta/coding-interviews?page=1

CCF-CSP：http://www.cspro.org/lead/application/ccf/login.jsp

\*注：本课程作业报告每周提交一次，请所有同学每周日晚21:00前交给各班学习委员，由学习委员统计并上传至微云，课堂上将随机抽取同学讲解自己的练习（5分钟）。

## 一、练习题目及代码

#### 【题目1】LeetCode27

给定一个数组 nums 和一个值 val，你需要原地移除所有数值等于 val 的元素，返回移除后数组的新长度。

不要使用额外的数组空间，你必须在原地修改输入数组并在使用 O(1) 额外空间的条件下完成。

元素的顺序可以改变。你不需要考虑数组中超出新长度后面的元素。

**【代码】**

**public class** Solution27 {

**public int** removeElement(**int**[] nums,**int** val){

**int** len=nums.**length**;

**int** pos=len-1;

**for**(**int** i=0;i<len;i++){

**if**(nums[i]==val){

**while** (pos>=0&&nums[pos]==val){*//不能反*

pos-=1;

}

**if**(pos>i){*//需要判断*

nums[i]=nums[pos];

pos-=1;

}

**else** {

**break**;

}

}

}

**return** pos+1;

}

}

**【提交记录截图】**



**【遇到的问题及解决办法】**

思路：pos指向数组最后一位，遍历一遍数组，如果遇到值等于val的元素，将数组中最后一位不等于val的值赋给该元素，pos移动，当pos移动到遍历元素时退出遍历；遇到的问题：刚开始判断条件写反了，报**java.lang.ArrayIndexOutOfBoundsException: -1**错误。

**【对比学习心得】**

#### 【题目2】LeetCode150

根据逆波兰表示法，求表达式的值。

有效的运算符包括 +, -, \*, / 。每个运算对象可以是整数，也可以是另一个逆波兰表达式。

说明：

整数除法只保留整数部分。

给定逆波兰表达式总是有效的。换句话说，表达式总会得出有效数值且不存在除数为 0 的情况。

**【代码】**

**import** java.util.Stack;

**import** java.lang.Integer;

**public class** Solution150 {

**public int** evalRPN(String[] tokens){

Stack<Integer> s=**new** Stack<Integer>();

**int** len=tokens.**length**;

**for**(**int** i=0;i<len;i++){

**if** (!isF(tokens[i])){

s.push(Integer.*valueOf*(tokens[i]));

}**else** {

**int** b=s.pop().intValue();

**int** a=s.pop().intValue();

**int** res=0;

**if**(tokens[i].equals(**"+"**)){res=a+b;}

**if**(tokens[i].equals(**"-"**)){res=a-b;}

**if**(tokens[i].equals(**"\*"**)){res=a\*b;}

**if**(tokens[i].equals(**"/"**)){res=a/b;}

s.push(res);

}

}

**return** s.pop();

}

**public boolean** isF(String str){

**if**(str.equals(**"+"**)||str.equals(**"-"**)||str.equals(**"\*"**)||str.equals(**"/"**)){

**return true**;

}**else return false**;

}

}

**【提交记录截图】**



**【遇到的问题及解决办法】**

思路：利用上学期数据结构中的后缀表达式求值的方法，遍历表达式，遇到数压入栈，遇到运算符取出栈顶两个元素进行计算，遍历结束时栈顶元素即为要求的值。

**【对比学习心得】**

#### 【题目3】LeetCode20

给定一个只包括 '('，')'，'{'，'}'，'['，']' 的字符串，判断字符串是否有效。

有效字符串需满足：

左括号必须用相同类型的右括号闭合。

左括号必须以正确的顺序闭合。

注意空字符串可被认为是有效字符串。

**【代码】**

**import** java.util.Stack;

**public class** Solution20 {

**public boolean** isValid(String s) {

Stack<Character> stack=**new** Stack<Character>();

**int** len=s.length();

**for**(**int** i=0;i<len;i++){

**if**(stack.empty()){stack.push(s.charAt(i));}

**else if** (s.charAt(i)==**')'**){

**if**(stack.pop()!=**'('**)**return false**;

}

**else if** (s.charAt(i)==**'}'**){

**if**(stack.pop()!=**'{'**)**return false**;

}

**else if** (s.charAt(i)==**']'**){

**if**(stack.pop()!=**'['**)**return false**;

}

**else** stack.push(s.charAt(i));

}

**if**(stack.empty())**return true**;

**else return false**;

}

}

**【提交记录截图】**



**【遇到的问题及解决办法】**

栈的基础用法，遍历字符串，遇到左括号压入栈，右括号匹配栈顶元素，若不匹配则括号无效，若匹配栈顶元素出栈，遍历完成栈空则字符串有效。

**【对比学习心得】**

#### 【题目4】LeetCode283

给定一个数组 nums，编写一个函数将所有 0 移动到数组的末尾，同时保持非零元素的相对顺序。

**【代码】**

**public class** Solution283 {

**public void** moveZeroes(**int**[] nums) {

**int** len=nums.**length**;

**int** i;

**for**(i=0;i<len;i++){

**if**(nums[i]==0)**break**;

}

**int** j=i;

**while** (i<len){

**if** (nums[i]!=0){

nums[j++]=i;

}

i++;

}

**while** (j<len){

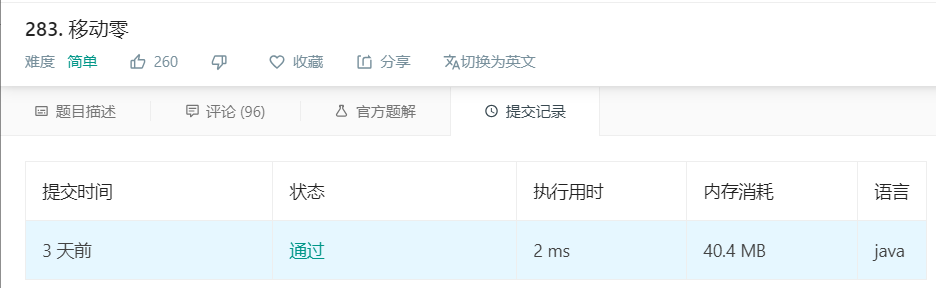
nums[j++]=0;

}

}

}

**【提交记录截图】**



**【遇到的问题及解决办法】**

遍历数组，得到第一个0元素的位置，再依次遍历之后的元素，若不为0则移动到第一个0元素的位置。

**【对比学习心得】**

#### 【题目5】LeetCode19

给定一个链表，删除链表的倒数第 n 个节点，并且返回链表的头结点。

**【代码】**

**class** Solution19 {

**public** ListNode removeNthFromEnd(ListNode head, **int** n) {

ListNode p1=head;

**for** (**int** i=0;i<n;i++){

p1=p1.**next**;

}

ListNode p2=head;

**if** (p1!= **null**){

p1=p1.**next**;

}**else** {

**return** head.**next**;

}

**while** (p1!=**null**){

p1=p1.**next**;

p2=p2.**next**;

}

p2.**next**=p2.**next**.**next**;

**return** head;

}

}

**【提交记录截图】**

**【遇到的问题及解决办法】**

两个指针找到要删除节点的前一个结点，删除该节点之后的节点；遇到的问题：一些特殊用例无法通过：①只有一个节点的链表②要删除的节点为头结点；加了一些判断条件之后通过。

**【对比学习心得】**

#### 【题目6】牛客网-二维数组中的查找

在一个二维数组中（每个一维数组的长度相同），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。

**【代码】**

**public class** nSolution1 {

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

**int** col=array[0].**length**-1;

**int** row=0;

**while** (row<array.**length**&&col>=0){

**int** temp=array[row][col];

**if** (temp==target)**return true**;

**else if** (temp>target)col-=1;

**else** row+=1;

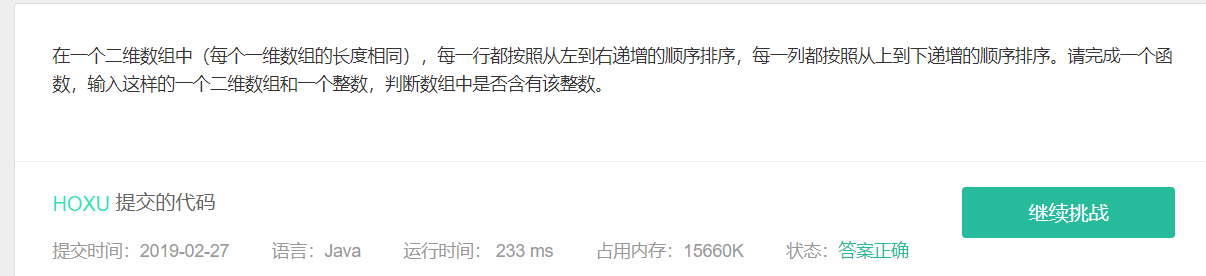
}

**return false**;

}

}

**【提交记录截图】**



**【遇到的问题及解决办法】**

上课说过方法，依次比较矩阵右上角元素，用代码实现了一下。

#### 【题目7】牛客网-替换空格

请实现一个函数，将一个字符串中的每个空格替换成“%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。

**【代码】**

**public class** nSolution2 {

**public** String replaceSpace(StringBuffer str) {

**int** len=str.length();

**int** amount=0;

**for**(**int** i=0;i<len;i++){

**if** (str.charAt(i)==**' '**)amount+=1;

}

**int** new\_len=len+2\*amount;

str.setLength(new\_len);

**int** i=new\_len-1;

**while** (i>=0){

**char** ch=str.charAt(len-1);

**if**(ch==**' '**){str.replace(i-2,i+1,**"%20"**);i-=3;len--;}

**else** {str.setCharAt(i,ch);i--;len--;}

}

**return** str.toString();

}

}

**【提交记录截图】**



**【遇到的问题及解决办法】**

遍历一遍字符串，统计空格数量，计算出替换之后的字符串长度后更改字符串长度，然后从后往前遍历字符串，依次移动替换元素即可。

#### 【题目8】牛客网-链表中倒数第k个结点

输入一个链表，输出该链表中倒数第k个结点。

【代码】

**public class** nSolution15 {

**public** ListNode FindKthToTail(ListNode head,**int** k) {

**if**(head==**null**)**return** head;

ListNode p1=head;

**int** count=0;

**int** i=k;

**while** (k--!=0&&p1!=**null**){

p1=p1.**next**;

count++;

}

ListNode p2=head;

**while** (p1!=**null**){

p1=p1.**next**;

p2=p2.**next**;count++;

}

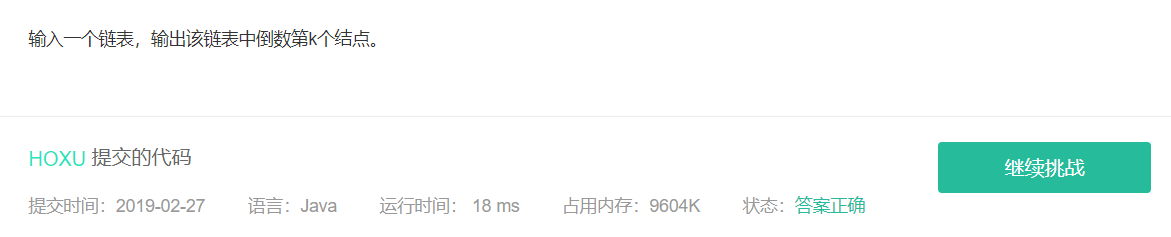
**if**(count<i)**return null**;*//之前没有保存k，k在上面被修改未发现*

**return** p2;

}

}

**【提交记录截图】**



**【遇到的问题及解决办法】**

与LeetCode19类似，用双指针找出倒数第n个节点。

## 二、本周小结

复习了上学期的一下学习的数据结构的用法。

-------------------------------

成绩评定：

教师签名：

批改日期：