湖南科技大学2017-2018第二学期

数据结构课程设计报告

**学 院** ： 计算机科学与工程

**专业班级：** 17软件工程一班

**姓 名：** 刘玉婷

**学 号：** 1705050131

**指导教师：** 符开耀

**时 间**： 2018.6.25-2018.7.6

**地 点**： 逸夫楼 机房

|  |
| --- |
| 指导教师评语：  **成绩：**  **签名：**  **年 月 日** |

# 任务一 复杂度分析1

## 任务描述

## 分析含有三个for循环的代码.问printf语句共执行了几次？这段代码执行完以后i+j+k值为多

## 二、分析与设计

（1）解决该问题有两种思路。一是直接暴利，设S++，直接输出S。这样很可能会超时。

二是分析三个for循环的结构，找出其中的规律，该方法在找规律时花费些许时间但超时的可能性几乎为0.

（2）具体的实现就是找出当n=4时，s=1；n=5，s=1+2，依次。运用一个for循环即可实现。

（3）具体流程图：

## 三、性能分析

算法的时间复杂度O（1）和空间复杂度O（1）分析。

## 四、调试与测试数据

调试程序时测了多组数据，n=3，4，5，6，7，10时s对应的取值。

## 五、小结

这题错了两次，主要原因是做题的时候太心急了，没有静下心来慢慢找规律。

# 任务二 复杂度分析2

## 任务描述

## 分析while循环的代码.问printf语句共执行了几次？这段代码执行完以后i+j+k值为多

## 二、分析与设计

（1）解决该问题有两种思路。一是直接暴利，设S++，直接输出S。这样很可能会超时。

二是分析三个for循环的结构，找出其中的规律，该方法在找规律时花费些许时间但超时的可能性几乎为0.

（2）具体的实现找规律，通过暴力找出了几组数据，发现与第一题的规律类似，于是便对第一题的代码稍加修改，一下就过了。

（3）具体流程图：

## 三、性能分析

算法的时间复杂度O（n\*n）和空间复杂度O（1）。

## 四、调试与测试数据

调试程序时测了多组数据，n=3，4，5，6，7，10时s对应的取值，并发现存在与第一题类似的规律，仔细对照两题的数据发现只有改一下n的值便能得到正确答案。之后一直是格式错误的提交。这题简单但是仍然错了三次，第一次是因为调试时写的printf函数忘了注释，第二次是以为只有这一个错误便直接改了然后提交，结果发现当n=0时的格式错误，于是又得多提交一次。

## 五、小结

这题错了三次，这个故事告诉我们一个道理，提交的时候要留意一下答案错误的提醒，可能还会有别的错误。

# 任务三 Josephus问题系列

## 任务描述

## n个人排成一圈，按顺时针方向依次编号1，2，3…n。从编号为1的人开始顺时针"一二"报数，报到2的人退出圈子。这样不断循环下去，圈子里的人将不断减少。最终一定会剩下一个人。试问最后剩下的人的编号。要求程序模拟题意来实现。

## 二、分析与设计

（1）解决该问题的三种进阶。随着数据越来越大，解题的办法也越来越趋于技巧化。第一题是直接暴利模拟，因为数据较小就过了。第二题是通过第一题所测的数据找出规律，运用公式也一下过了。第三题随着数据越来越大，输出的结果越来越多样，解答的过程异常漫长。

（2）第一题模拟，第二题打表。第三题运用公式。

（3）这三题的具体流程图分别为：

## 三、性能分析

算法的时间复杂度O（n）和空间复杂度O（1）。

## 四、调试与测试数据

调试程序时测了多组数据，n=3，4，5，6，7，10，11，12，13~35时s对应的取值。从中寻找规律。

## 五、小结

编写程序时出了很多的小bug，首先我的代码每次都是挤在一块写，没有写分函数的习惯导致了调试时增加了很多可以避免的困难。其次是我在命名变量是总是贪图省事而命名一些相似的便量如i，ii，等，增加了我编写代码时的出错几率。两个很差的坏习惯导致我总是要比别人晚一点ac.

# 任务七 交集

## 任务描述

## 有两个相等长度的正整数序列A和B，都是有序的(递增排序)，同时一个序列中没有重复元素，现在需要求这两个序列的交――序列C，同时打印输出。

## 二、分析与设计

（1）解决该问题思路为设置两个变量i，j依次对应的边查找边比较。

（2）具体的实现就是若前一个数组的当前元素与后一个数组不相等且小于后一个数组则前一个数组向后查找，反之，则后一个数组向后查找。当两个查找数组有一个到达末尾时，查找结束。

（3）具体流程图：

## 三、性能分析

算法的时间复杂度O（n）和空间复杂度O（n）分析。可以结合OJ的评测数据，但是OJ是黑盒测试，可以给出用自己的数据集测试的结果。

## 四、调试与测试数据

调试程序时测了多组数据，n=3，4，5，6，7，10时s对应的取值。

## 五、小结

该题比较熟悉，之前写过了几次。就是编写程序时写了一个小错误，花费了一会的时间才调试好。

# 任务八 大爱线性表

## 一、任务描述

# 给线性表增加两个特殊的操作，第一个是‘R’ 操作，表示逆转整个表，如果表长为Ｌ，原来的第ｉ个元素变成第Ｌ-i+1个元素。第二个操作是‘D’，表示删除表的第一个元素，如果表为空，则返回一个“error”信息。给定表和一个操作串S，求出执行S后的表。

## 二、分析与设计

（1）解决该问题有两种思路。一是运用线性表，模拟操作过程。二是用数组，通过头尾删除来达到和题目要求的效果。因为此题运用的数据很大，所以粗略判断用线性表超时的可能性极大。故选用方法二。

（2）具体的实现为：

1.先将表中的数据量n分为两类，n=0和n>=1.

2.先讨论n=1时的输入情况。通过观察得无法直接输入整数数组，于是要从输入的字符串中提取出数值。且由于字符串的性质，整数23在字符串中占了两个字符，故需将字符串分离划一。然后设初始表中的状态为k=0，通过判断k的奇偶性来区分1.最后的表是要正向输出还是逆向输出2.删除表中元素的头部还是尾部。通过计数变量t来记录D的个数，若t>n则操作错误，对应输出error。

3.若n=0，则直接检测操作串中是否出现D，若出现则对应输出errror，否则输出空串。

（3）具体流程图：

## 三、性能分析

算法的时间复杂度O（n）和空间复杂度O（n）分析。可以结合OJ的评测数据，但是OJ是黑盒测试，可以给出用自己的数据集测试的结果。

## 四、调试与测试数据

调试程序时通过提供的样例数据来调试，一开始不知道scanf函数无法直接输入%d，等格式。写出错误代码，运行时的结果错得离谱。后来通过尝试发现需通过字符串来提取表中的数值。之后一直在完善代码。由于不知道如何初始化字符串，浪费了很长的一段时间。最后找到错误提交了俩遍，原因是字符串开得太小，还有就是代码中的strlen函数占用了太多的时间。将strlen函数省掉改成a[i]！=‘\0‘节省了很多时间，成功ac了。

## 五、小结

做这题的时候缺乏自信心，在调试程序时发现结果错得离谱，没有坚持继续调试而是认为自己做不出来放弃了。通过这一题，对scanf函数和字符串初始化有了更深的理解。这个故事告诉我们一个道理，ac一定有方法。另外，我的代码能力很差，有待加强，主要是敲的速度太慢了。

# 任务九 单词检查系列

## 一、任务描述

字典由下面几个单词组成：

bake cake main rain vase

如果输入文件中有词vake ，检查程序就能发现其是一个错误的单词，并且给出 bake, cake或vase做为修改建议单词。

## 二、分析与设计

（1）解决该问题思路为设置两个结构体组分别储存字典里的单词和单词长度以及查找的单词及单词长度，之后再进行比较。

（2）具体的实现为，由题意可得当单词在字典中找不到时找出相似的单词输出。相似的单词允许与查找的单词有一次不同。在此做标记即可。

（3）题一题二的具体流程图分别为：

## 三、性能分析

算法的时间复杂度O（n\*n）和空间复杂度O（n）分析。可以结合OJ的评测数据，但是OJ是黑盒测试，可以给出用自己的数据集测试的结果。

## 四、调试与测试数据

调试程序时测了多组数据，n=3，4，5，6，7，10时s对应的取值。

## 五、小结

题一刚开始以为顺序表就是单向链表在单向链表里浪费了很多时间，后来又因为单向链表中无法存入字符串而受阻。最后受到启发才想起结构体来。

题二一开始不会创建树的结构体指针。后来在班上李正恒大佬的指导下顿悟，真的是蛮开心。题一的单向链表磨了近一天，终于解决了。有时候请教大佬（如果大佬愿意教的话）比自己看书懂得更快。

# 任务十二 后缀表达式求值

## 任务描述

## 扫描后缀表达式，凡遇操作数则将之压进堆栈，遇运算符则从堆栈中弹出两个操作数进行该运算，将运算结果压栈，然后继续扫描，直到后缀表达式被扫描完毕为止，此时栈底元素即为该后缀表达式的值。

## 二、分析与设计

（1）解决该问题思路为按要求暴力模拟。

（2）具体的实现为，由题意构建栈，并且写出出栈和入栈的分函数。并且要写出一个函数来将字符串转换为数字。

（3）具体流程图分别为：

## 三、性能分析

算法的时间复杂度O（n）和空间复杂度O（n）分析。可以结合OJ的评测数据，但是OJ是黑盒测试，可以给出用自己的数据集测试的结果。

## 四、调试与测试数据

调试程序时用了样例，主要是调试栈的输出。后来发现问题出在出栈后的减数与被减数问题弄错了。先出栈的数据应该是被减数。

1. **小结**

通过这道题我学会了简介别人的一小段代码或者思路来解决自己的问题，这题的花了一早上才做出来，主要原因还是我不够自信，另外就是我吸收知识的水平有所欠缺，觉得自己不行，所以越做越没有信心，越做越急躁。后来午饭回来换了个思路就调试成功了。之前一直在做无用功。

# 任务十三中缀转化为后缀表达式

## 任务描述

## 输入一个中缀表达式，编程输出其后缀表达式，中缀表达式转后缀表达式的方法：1.遇到操作数：直接输出（添加到后缀表达式中）2.栈为空时，遇到运算符，直接入栈3.遇到左括号：将其入栈4.遇到右括号：执行出栈操作，并将出栈的元素输出，直到弹出栈的是左括号，括号不输出5.遇到其他运算符：加减乘除：弹出所有优先级大于或者等于该运算符的栈顶元素，然后将该运算符入栈6.最终将栈中的元素依次出栈，输出。

## 二、分析与设计

（1）解决该问题思路为按要求暴力模拟。

（2）具体的实现有两种，一种为构建栈且写出出栈和入栈的分函数。难点为不好实现操作4和操作五。尝试了一下后编不出来便开始思考第二种思路。

第二种思路比较简单，题目只输入一个字符串，我们可以再构建一个字符串模拟栈。运用一个分函数来判断运算符的优先级。并且给已经输出的运算符做标记防止它重复输入。最后遍历一遍模拟数组检查是否还有没有输出的运算符。

（3）具体流程图分别为：

## 三、性能分析

算法的时间复杂度O（n）和空间复杂度O（n）分析。可以结合OJ的评测数据，但是OJ是黑盒测试，可以给出用自己的数据集测试的结果。

## 四、调试与测试数据

调试程序时用了样例，主要是还是输出问题。

1. **小结**

脑子有点蒙，越写低级的bug就越多。通过这题得知自己调试程序时的速度不够。如果每回写完代码运行前自己有先检查一遍的习惯而不是一味依靠编程软件报错应该会好一些。毕竟有一些逻辑错误软件是很难找到的。