|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《数据结构》实验报告** |
|  |
| 实验二  栈与队列的应用  学 院: 计算机科学与技术   |  |  | | --- | --- | | 姓 名: | 朱海峰 | | 学 号: | 190110716 | | 专 业: | 计算机科学与技术 | | 日 期: | 2020-04-11 | |

# 一、问题分析

原问题分为实验内容一和实验内容二。

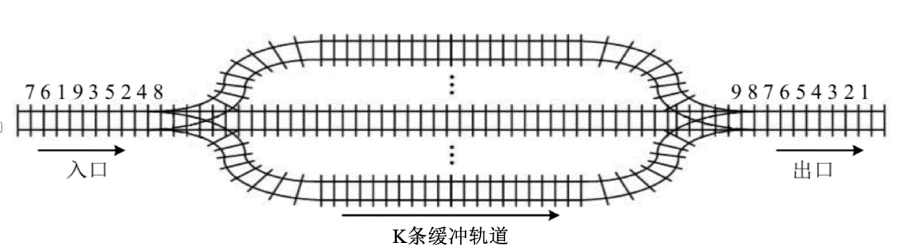
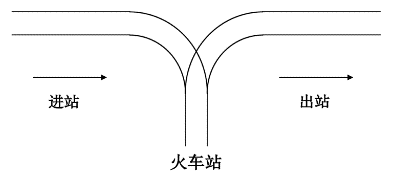
其中实验内容一：如下图1所示，某火车站只可单方向进出站。假设n列火车按照给定顺序进站，请判断是否可以按给定顺序出站。要求用栈的方法实现。

图 1 图 2

实验内容二：如上图2所示，入口到出口之间有*k*条缓冲轨道，火车的行进方向均为从左至右，可驶入任意一条缓冲轨道。现有编号为1~*n*的*n*列火车乱序驶入，若期望驶出的次序依次为1至*n*，请输出*k*的最小值。要求用队列的方法实现。

分析如下：

实验内容一中，中间的轨道用栈来模拟；实验内容二中，缓冲轨道用多个队列来模拟。给出的火车进出序列直接用两个整型数组来表示。

火车依次进出可用压栈、出栈与入队、出队模拟，从而将火车问题转化为计算机解决的问题。

# 二、详细设计

## 2.1 设计思想

按照火车顺序，即数组顺序，在实验内容一中通过压栈，循环判断栈顶是否符合出站序列，若符合则开出一节火车，即出栈，不符合则开入一节火车，即压栈，最后完成则说明可行，否则不行。

在实验内容二中，依次将符合次序火车开入同一队列，均不符合的开入新的队列，最终用到的队列数即*k*的最小值。

## 2.2 存储结构及操作

(1) 存储结构

实验内容一中的轨道用链栈模拟，实验内容二中的多条缓冲轨道用以指针数组指向的多个链式队列来模拟。另外，火车的出入序列简单地用两个整型数组表示即可。

(2) 涉及的操作

LinkStack\* InitStack()

/\*\* 初始化空栈并返回 \*/

int Push(LinkStack \*linkStack,int ele)

/\*\* 为ele分配节点内存并压入栈，操作成功返回1 \*/

int Pop(LinkStack\* linkStack)

/\*\* 将一个节点弹出栈，操作成功返回1 \*/

int GetTop(LinkStack\* linkStack)

/\*\* 若栈不空，返回栈的栈顶元素 \*/

int EnQueue(LinkQueue \*linkQueue,int ele)

/\*\* 为ele分配节点内存并入队 \*/

int findClosestQueue(LinkQueue \*railQueue[],int usedQueue,int curTrain)

/\*\* 为curTrain在railQueue中找到最合适的火车轨道，未找到返回-1 \*/

## 2.3 程序整体流程

(1) 整体流程，如下图3；

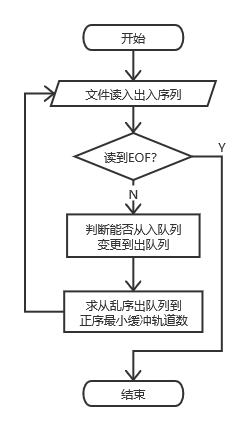


图 3 图4

(2) 实验内容一，判断能否从入序列变更到出序列，如上图4；

(3) 实验内容二，求乱序出队列变更到正序，如下图5；

(4) 实验内容二中，为当前火车找到合适已用队列，如下图6。

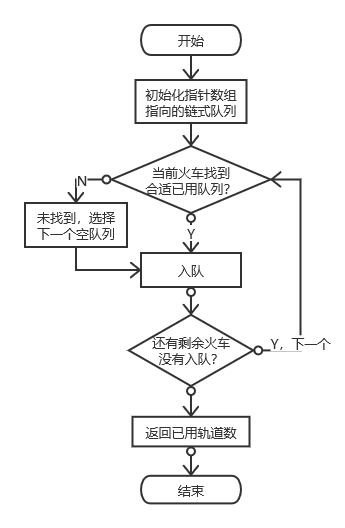
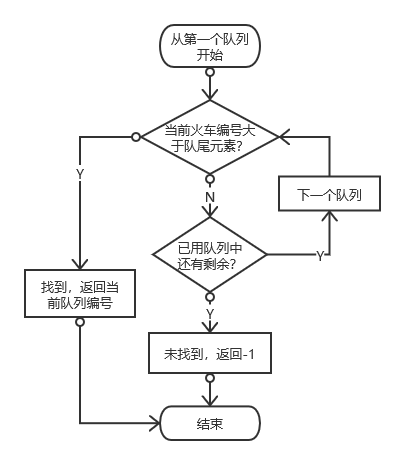
 

图 5 图 6

# 三、用户手册

此程序在文件中输入数据，在main函数中while条件控制处可更改读入文件，程序默认从TestTrain中读入数据，您也可以直接在同目录下的TestTrain.in中输入您想测试的数据。

程序需要依次输入：该情况的火车总节数、进站序列和出站序列，如：

5

1 2 3 4 5

3 4 5 1 2

当然也可以批量输入，如：

8

1 2 3 4 5 6 7 8

4 6 5 3 2 8 7 1

8

1 2 3 4 5 6 7 8

3 2 1 4 8 7 5 6

9

1 2 3 4 5 6 7 8 9

4 6 5 7 3 2 1 9 8

输入后即可开始编译运行，无需其他操作。对应每个情况，控制台会输出对应的从入站序列到出站序列变更是否可行、从出站乱序到正序需要最少的缓冲车道。

感谢您的使用！

# 四、结果

对测试文件TestTrain.in的运行结果如下图7、图8。

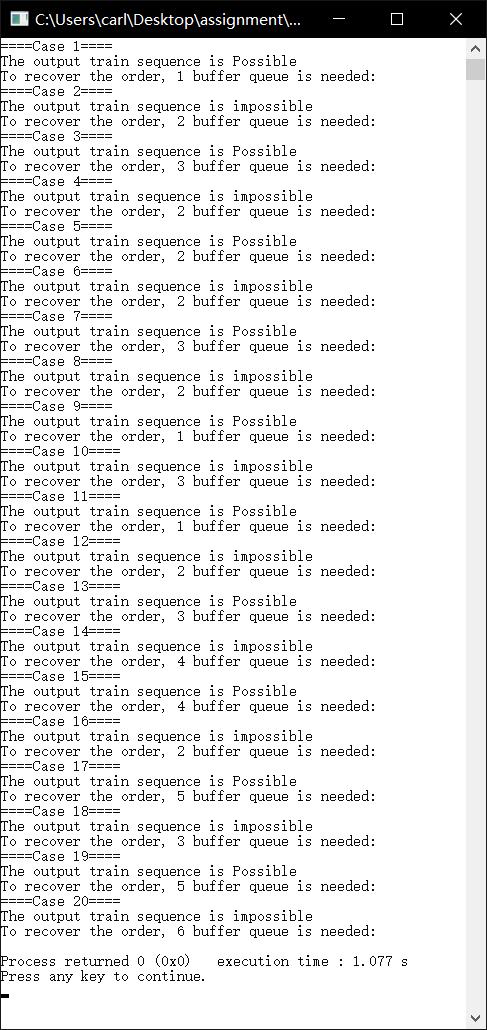


图 7

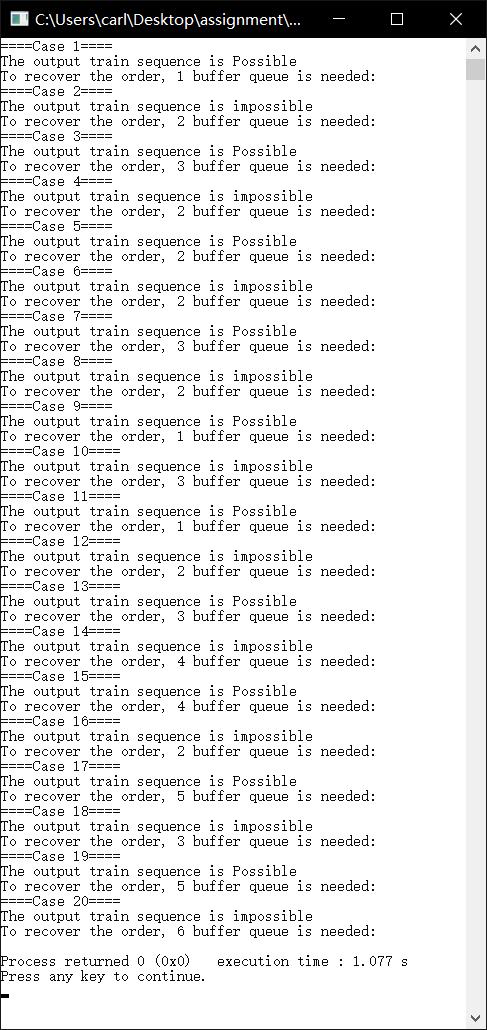


图 8

# 五、总结

此次实验中，让我体会到的正是“大行细谨起”。实验内容总体并不难，但是过程中却很曲折，我列举一下遇到的坎坷。

我解决了一些问题：

实验内容一很快就写完了，但是找一个bug找了一个小时，最后发现是栈的栈顶元素获取由于复制了部分弹出函数的代码，没注意到区别出了问题。函数还是要一行一行的写，复制的话也要每行来检查。

写了一点实验内容二，没写完，再运行想检查实验内容一，程序崩了。老是在纠结明明没改为什么运行不了了，奇怪的是单行调试竟然可以坚持着把20个测试完成。也是想了很久才发现是新写的实验内容二影响了，我直接把调用实验内容2部分注释了，才完成课上实验截图。

我遇到了一些困难：

当我写完实验内容二，运行结果不对，想单行调试，但是每次到了while的条件，即scanf哪里，debugger无提示停止。我以为是代码问题，因为之前的单步调试都没问题，就把老师发的模板main.c复制进工程文件拿来单步调，竟然出现同样问题。而我在实验内容一单步调试时经过此处没有任何问题，据此，我怀疑debugger坏了，我决定卸载重装codeblocks，最新版的debugger用不了，提示是找不到函数边界，百度无果；重装17版codeblocks，问题不变。如此反反复复，重装了几遍后，我用每步打印、肉眼看的办法，终于找到bug，运行没问题单行调试还是老问题，现在也没有解决。

这个问题我还是怀疑是代码的问题，因为别的程序单步调试也没问题。路径无中文，工程文件，可能是内存访问问题（因为scanf后相关变量未赋值，但是我反复检查和老师的模板代码没区别，没改过），但是无提示。我害怕以后再出现这种情况。

总结一下：

1. 代码要一行一行码，复制可能有问题；
2. 从可运行到不可运行，问题涉及在刚刚改了的地方，不一定就是，但是有关；
3. 崩溃位置的打印内容附近可能是bug，附近每个函数都要into+单行；
4. 有问题先放一放，解决不了先睡一觉，精神好了再看。