湖南大學

数据结构

课程实验报告

题　　目　学生排队（CCF201703-2）

学生姓名　　李晓彤

学生学号　　201726010128

专业班级　　软件工程1701

完成日 期　　2018.10.09

一、需求分析

1.问题描述

体育老师小明要将自己班上的学生按顺序排队。他首先让学生按学号从小到大的顺序排成一排，学号小的排在前面，然后进行多次调整。一次调整小明可能让一位同学出队，向前或者向后移动一段距离后再插入队列。

例如，下面给出了一组移动的例子，例子中学生的人数为8人。

0）初始队列中学生的学号依次为1，2，3，4，5，6，7，8；

1）第一次调整，命令为“3号同学向后移动2”，表示3号同学出队，向后移动2名同学的距离，再插入到队列中，新队列中学生的学号依次为1，2，4，5，3，6，7，8；

2）第二次调整，命令为“8号同学向前移动3”，表示8号同学出队，向前移动3名同学的距离，再插入到队列中，新队列中学生的学号依次为1，2，4，5，8，3，6，7；

3）第三次调整，命令为“3号同学向前移动2”，表示3号同学出队，向前移动2名同学的距离，再插入到队列中，新队列中学生的学号依次为1，2，4，3，5，8，6，7。

小明记录了所有调整的过程，请问，最终从前向后所有学生的学号依次是多少？

请特别注意，上述移动过程中所涉及的号码指的是学号，而不是在队伍中的位置。在向后移动时，移动的距离不超过对应同学后面的人数，如果向后移动的距离正好等于对应同学后面的人数则该同学会移动到队列的最后面。在向前移动时，移动的距离不超过对应同学前面的人数，如果向前移动的距离正好等于对应同学前面的人数则该同学会移动到队列的最前面。

2.输入数据

输入的第一行包含一个整数n，表示学生的数量，学生的学号由1到n编号。

第二行包含一个整数m，表示调整的次数。

接下来m行，每行两个整数p，q，如果q为正，表示学号为p的同学向后移动q，如果q为负，表示学号为p的同学向前移动-q。

3.输出数据

输出一行，包含n个整数，相邻两个整数之间由一个空格分隔，表示最终从前向后所有学生的学号。

4.测试样例设计

样例输入一

```
8
3
3 2
8 -3
3 -2
```

样例输出一

```
1 2 4 3 5 8 6 7
```

说明：三号同学向后移动两位，然后八号同学向前移动三位，最后三号同学再向前移动两位。

样例输入二

```
5
2
1 4
1 -4
```

样例输出二

1 2 3 4 5

说明：一号同学向后移动四位再向前移动四位。

样例输入三

10

3

5 5

10 -3

9 1

样例输出三

1 2 3 4 6 10 7 8 5 9

说明：五号同学向后移动五位，然后十号同学向前移动三位，最后九号同学向后移动一位。

样例输入四

3

1

1 2

样例输出四

2 3 1

说明：一号同学向后移动两位。

样例输入五

8

4

3 2

8 -3

3 -2

2 6

样例输出五

1 4 3 5 8 6 7 2

说明：三号同学向后移动两位，然后八号同学向前移动三位，然后三号同学再向前移动两位，最后二号同学再向后移动六位。

二、概要设计

1.抽象数据类型

为实现上述程序的功能，可以用数组存储队列。并将用户输入的值用for循环来实现一次输入一次移动的调整队列模拟。抽象数据类型设计：

数据对象：一组整数

数据关系：一组整数通过for循环依次存入，按照存储的先后次序，满足线性特征，即<ai,ai+1>(0≤a＜n)。

基本操作：准备一个能够存储一组整数的存储空间 判断当前输入的整数是否已经保存若未保存，保存该整数。

ADT IntegerSet {

数据对象：D = { ai | ai Î 整数，i = 1,2，…, n, n>=0}

数据关系：R = {< ai-1，ai> | ai-1, ai ÎD}

基本操作：

void init ( )//操作功能：初始化，构造出一个空表。

void setstart ( )//操作功能：将当前位置定位到表头。

void next ( )//操作功能：将当前位置移动到下一个元素。

bool getvalue (const Elem&)
//操作功能：获取当前位置的元素值，赋值给 Elem。
//若成功，返回真，否则返回假。
bool insert (const Elem&)
//操作功能：将元素 Elem 插入到线性表的当前位置。
//若插入成功，返回真，否则返回假。

}

2.算法的基本思想

排队问题中的数据是一个个整数， 而这种数据是存在“第一元素、最后元素”， 并且存在“唯一的前驱和后继的”，符合线性表的特点。由于需要统计各个值的位置并移动它到指定位置，可以采用顺序表来实现线性表，完成最终队列顺序的输出。

核心算法主要分为两步：

1、确定要移动的序号的位置。

2、按要求移动该学生。

已知输入的学生数量n，我们可以用数组存储这个人数对应的队列。然后利用循环来实现队列的移动。先从键盘中读取用户输入的需要移动的学生序号和向前或向后移动个数，然后确定该队列中该学生的位置，并且用变量来存储该位置，这里完成了算法第一步。算法第二步就是开始移动该学生，先判断是向前移动还是向后移动，若向前移动，则该学生后面的学生开始往前移动，并占据他的位置，然后再后面的学生往前移动占据前一位学生的位置，直到满足用户输入的需要移动的位数，然后学生停止往前移动，此时将该学生插入到已经移动的最后一个学生的后面完成移动。向前移动则类似，只是要让需要移动的学生的前面的学生开始往后移动位置，移动操作都和向后移动的操作一致。然后重复此操作直到完成用户需要的移动次数为止。

3.程序的流程

程序由三个模块组成：

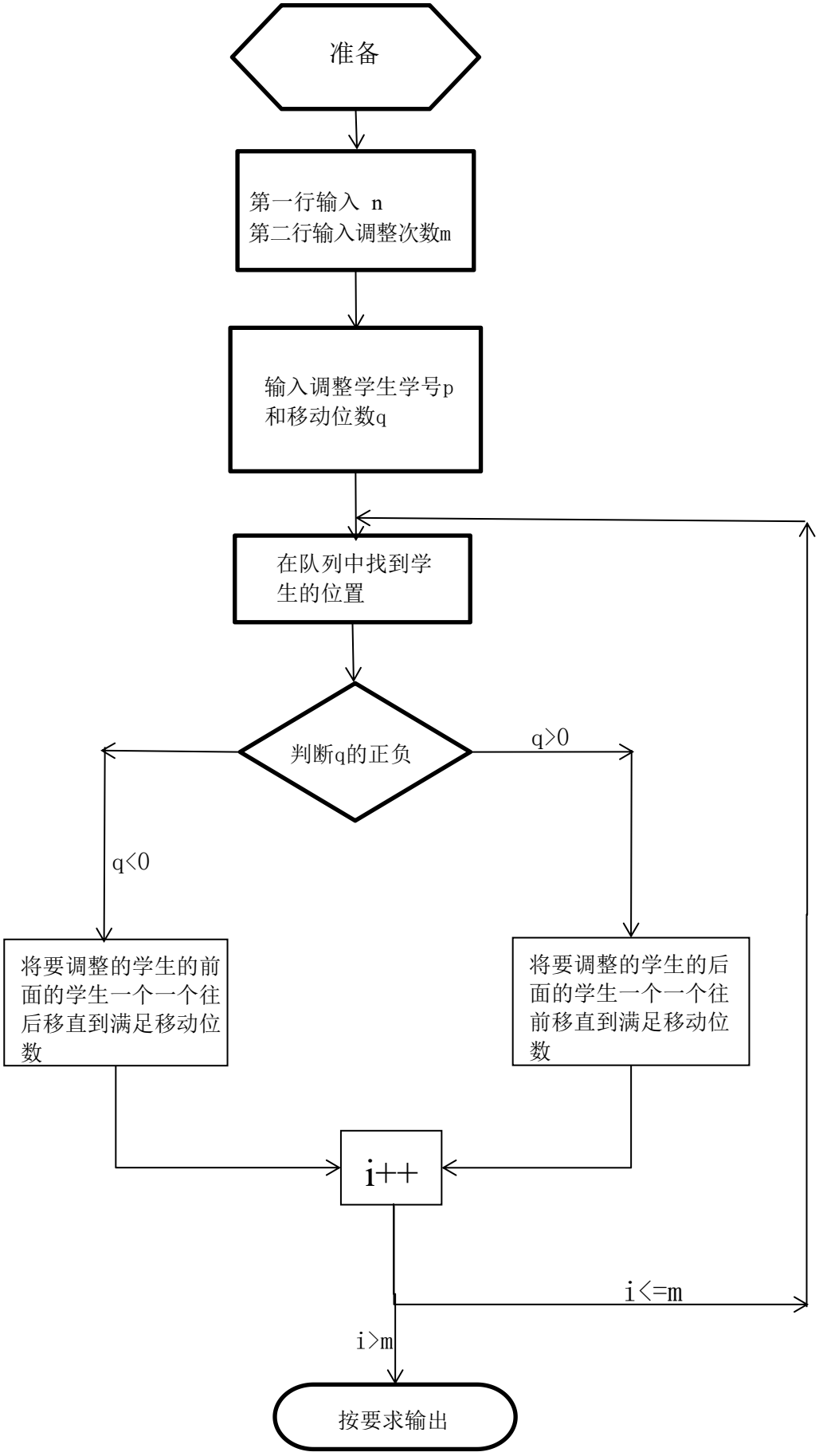
（1）输入模块：无提示语句，第一行输入学生总数 n，第二行输入调整次数m，接下来的m行输入要调整的学生序号p和移动位数q，中间用空格隔开。

（2）处理模块：将队列的学生序号储存于顺序表中。在顺序表中查找要调整的学生序号的位置，然后保存下来，用if语句来判断向前移动还是向后移动。然后用for循环来移动后面的学生，移动完之后再插入调整的学生序号，这样就能完成学生移动。

（3）输出模块：在 DOS 下，按要求输出最终从前向后所有学生的学号。

流程图如下：

流程图：


准备
第一行输入 n
第二行输入调整次数m
输入调整学生学号p
和移动位数q
在队列中找到学
生的位置
判断q的正负
q>0
q<0
将要调整的学生的前面的学生一个一个往后移直到满足移动位数
将要调整的学生的后面的学生一个一个往前移直到满足移动位数
i++
i<=m
i>m
按要求输出

三、详细设计

1.物理数据类型

由于输入数据是整数，且规定评测用例，1 <= n <=1000,1 <= m <= 1000,所有移动均合法。所以数据类型用 int。由于一组整数由for循环依次存入，按照存入的先后次序，满足线性特征，所以物理数据结构为线性结构。

2.输入和输出的格式

从键盘输入数据规模和移动操作的具体内容，程序则将该组数据做模拟排队，最后输出排好的队列。

3.算法的具体步骤

一：设置数组，通过for 循环将数据存入

二：已知输入总数n，和操作总数m，还有每一步要移动的学生序号p和移动位数q。先在顺序表中确定该学生的位置，并且记下，然后看q的正负。如果q为正，则将要移动学生后面的学生挨个往前移动，直到移动的学生个数满足移动位数q。然后将要移动的学生p插入已经移动的学生的后面完成移动。如果q为负，则将要移动学生前面的学生挨个往后移动，直到移动的学生个数满足移动位数q。然后将要移动的学生p插入已经移动的学生的前面完成移动。反复进行上述确定位置和元素位置的操作，直到满足操作总数m。

三：

ADT 具体实现：

```
  const int N = 1000;
  int stu[N];
  cin>>n;
  for(int i=0;i<n;i++)
{
      stu[i]=i+1;
}
```

4.算法的时空分析

1、初始化部分为循环赋值，时间复杂度为 Θ(n)。

2、处理部分，我采用的是 for 循环查找学生p位置的方法，并且用for循环来移动学生。所以说最差情况是移动n次，最好情况是移动一次，故时间代价较高，时间复杂度为Θ(n^2)。

综上，该算法的时间代价为 Θ(n^2)。当移动次数较少时，时间复杂度为 Θ(n)。

(事实上要用链表来完成这道题，其时间复杂度则大大减小，利用链表的插入与删除优势则可减少操作部分的时间复杂度)

四、调试分析

1.调试方案设计

撰写调试计划和方案。包括：调试目的，样例，调式计划和设置。

调试目的：为了发现代码是否有语法错误、连接错误、逻辑错误和运算错误，有不严谨之处。

样例：

```
8
3
3 2
8 -3
```

3 -2

调试计划：设定好断点，单步执行，查看local、ylocal、stu数组的值的变化。寻找问题，检查遗漏。

设置：设置好断点应在哪一步，要有几个断点。

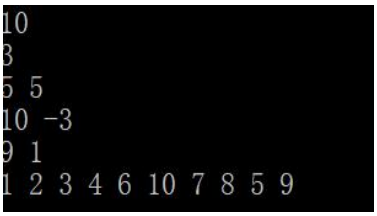
2.调试过程和结果，及分析

开始时未加入查找学生p位置的操作，导致移动的并不是该学生。然后加入了查找学生位置的操作并用local储存后发现，原先最后把p插入的位置是用local+q-1来计算要插入学生p的位置，但是在local累加或者递减的时候，local的值是改变的，这也就导致了最后计算插入学生p位置的计算错误，所以用ylocal存储计算前的位置，用local进行累加或者递减操作。经过再次调试后，输出结果正确，该代码提交得分为 100 分。

五、测试结果

三号同学向后移动两位，然后八号同学向前移动三位，最后三号同学再向前移动两位，结果正确

一号同学向后移动四位再向前移动四位结果正确



五号同学向后移动五位，然后十号同学向前移动三位，最后九号同学向后移动一位，结果正确。

一号同学向后移动两位，结果正确。

三号同学向后移动两位，然后八号同学向前移动三位，然后三号同学再向前移动两位，
最后二号同学再向后移动六位，结果正确。

六、实验心得

做实验 0，要求自己找一道 CCF 原题，我找的是 2017年3月考试的第二题。实验要求说要用到储存一个数据集，然后再处理。本来想用链表来处理这道题，但是链表不会写，没办法只能用数组来写这道题。也上网搜了一下链表解决的，觉得比起数组来说方便很多，今后打算遇到这种题尝试用链表解决一下。通过此次撰写实验报告 0，我能更加了解和掌握线性表的知识，并且了解到了许多数据结构方面的思想、思考方式。感觉学习数据结构后思考代码的方式都变得有所不同，编写代码的水平都有了提高。数据结构的学习还是很有必要的。