实验报告

-project1

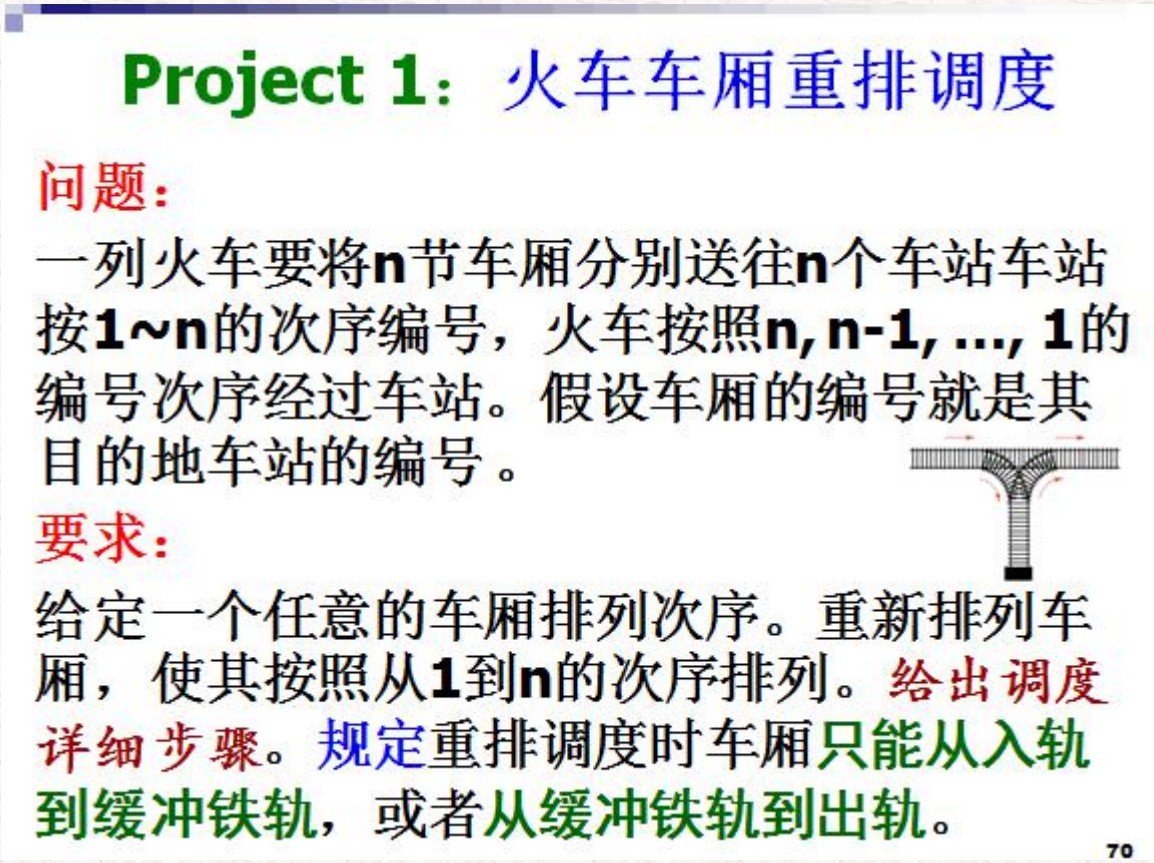
组长：王友坤
组员：王迎旭 王显淼
班级：16 级软件工程 6 班
日期：2017.9.24

目录

第一部分

问题描述与实验目的

问题描述：



实验目的：

将各节车厢按编号从大到小挂到车头上，其中在入轨与出轨之间有 k 条缓冲铁轨，将通过缓冲铁轨完成本次的火车车厢的重排。

第二部分

编译细节的介绍

2.1 运行环境

Windows10

2.2 编程语言

c++

2.3 开发工具

（1）Windows 环境下的 g++编译器与 Sublime Text3 编辑器

（2）Dev c++（注:由于本地的编写实在 Sublime 中完成，所以在 c++中编译时候会产生注释乱码情况）

2.4 编码的规范

在 Sub 中编辑时候，已经严格按照首行缩进的方式进行编译。但美中不足，在代码进行复制粘贴时候有可能导致代码错位的问题。

第三部分

分析与设计

3.1 数据结构

（1）栈
（2）数组

3.2 设计思路

（1）在 c++中实现对栈的基本操作，从而实现对火车车厢的调度重拍

（2）通过使用相应数据结构，对其进行一步一步的操作处理判断，使其完成相应功能

（3）输出相应的结果

3.3 设计要点

（1）数据结构的申请与构建

（2）各种出栈入栈条件的设计

（3）操作过程的监控

（4）前端显示的设计

（5）内存空间的检测

（6）非法数据的处理

3.4 设计规则

（1）车厢从入轨的前部（即右端）只可以移动到一个缓冲铁轨的顶端或出轨的右端。

（2）缓冲铁轨的顶端的车厢只可以移动到出轨的最左端

（3）车厢移动到出轨的最左端（即火车头端）后不能再移动

3.5 算法设计

（1）接收一组由 n 个数组成的数据，按倒序对数据进行处理。

（2）设置一个参照值，参照值代表下一个要过去的车厢编号。下一个过去的车厢可能在入轨队列中，也可能在缓冲铁轨中。若参照值与入轨队列队头或缓冲铁轨的顶部的车厢编号相等，则将该车厢出轨，并将参照值加一。

（3）若车厢编号不等于参照值，则将其放进缓冲铁轨中。所有的缓冲铁轨用栈来表示，栈储存在一个 vector 容器中，从 0 开始计数。从第 0 条缓冲铁轨开始，遍历所有的栈顶元素，若栈顶元素大于该车厢编号，便将该车厢压栈，否则检查下一个栈。

（4）当所有栈顶都遍历过但没有符合条件的栈时，新开一个栈，将该车厢压栈（相当于现实中有多条缓冲铁轨，需要用到多少条就申请多少条）。每次参照值变化，就要对比入轨队列队头和遍历所有栈的栈顶。直到所有车厢成功出轨。

第四部分

关键代码展示

（放置在代码包中）

第五部分

测试程序部分

```
please input the number of train:
6
input train number:
2 3 4 1 6 5
here will build a new stack and 2 will be pushed in!
here will build a new stack and 3 will be pushed in!
here will build a new stack and 4 will be pushed in!
1 will leave!
here will build a new stack and 6 will be pushed in!
2 will be poped from stack[1] and leave!
3 will be poped from stack[2] and leave!
4 will be poped from stack[3] and leave!
5 will leave!
6 will be poped from stack[4] and leave!
1 2 3 4 5 use 4 stack
请按任意键继续. . .
```

第六部分

项目总结

一.本次项目通过对火车调度这一实际问题的分析与设计，加深了小组成员对栈的认识与了解：栈是限定仅在表头进行插入和删除操作的线性表。它按照先进后出的原则存储数据，先进入的数据被压入栈底，最后的数据在栈顶，需要读数据的时候从栈顶开始弹出数据（最后一个数据被第一个读出来）。

二.解决实际问题的算法要具有以下的特征才能有效的完成设计要求：

（1）有穷性。算法在执行有限步后必须终止。

（2）确定性。算法的每一个步骤必须有确切的定义。

（3）可行性。满足实际问题的需求。

三.在算法设计的整个过程中，小组成员一共提出了两个模型。为排除极端数据对算法的影响，小组成员不断测试程序并修复算法中的不足之处。为适应实际问题的操作，小组成员不断测试程序并改进算法。

四..在小组实践过程中，需要发挥小组各成员的积极性和能动性，同时要发扬团队协作精神。

第七部分

项目分工

算法设计：王显淼；代码设计：王友坤；实验报告：王迎旭