**武汉大学计算机学院**

**本科生实验报告**

**数据结构实验报告**

**实验三：停车场管理程序**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：数据结构

指 导 教 师 ：安 扬

学 生 学 号 ：2017301500061

学 生 姓 名 ：彭 思 翔

学 生 班 级 ：计科二班

上 机 环 境 ：Visual Studio Code

二○一八 年 11 月

**一、实验题目**

实验三：停车场管理程序

【问题描述】

设停车场内只有一个可停放n辆汽车的狭长通道，且只有一个大门可供汽车进出。汽车在停车场内按车辆到达时间的先后顺序依次由南向北排列，若停车场已停满n辆车，则后来的汽车只能在门外的便道等候，一旦有车开走，则排在便道上的第一辆车即可开入；当停车场内某辆车要离开时，在它之后进入的车辆必须先退出停车场为它让路，待该辆车开出大门外，其他车辆再按原次序进入停车场，每辆停放在停车场的车在它离开时必须按停留的时间长短交纳费用。

**二、实验项目的目的**

深入掌握栈的应用和队列应用的算法设计。

1. **实验项目程序结构**

面向过程函数

Main函数

栈实现函数

队列实现函数

getTop

destroyStack

queueEmpty

getLenth

init

initQueue

initStack

main

destroyQueue

work

inQueue

stackFind

deQueue

stackEmpty

formatCar

pop

构造函数

push

**四、实验项目中各文件函数功能描述**

void init(); //读入停车场规模

void work(); //操作函数，进行停车取车等操作

Car formatCar(int, long long); //构造汽车类，第一个参数是id，第二个参数是绝对秒数

bool queueEmpty(Queue\*); //判断队列是否为空，1为空，0为不空

void initQueue(Queue\*&); //初始化队列

void destoryQueue(Queue\*&); //销毁队列

bool inQueue(Queue\*&, Car); //元素入队列

bool deQueue(Queue\*&, Car&); //元素出队列

bool pop(Stack\*&); //弹出栈顶元素

Car getTop(Stack\*&); //获取栈顶元素

int getLenth(Stack\*&); //获取栈长度

bool push(Stack\*&, Car); //入栈

void initStack(Stack\*&); //新建栈并初始化

bool stackEmpty(Stack\*&); //判断栈是否为空，为空返回1，不为空返回0

void destroyStack(Stack\*&); //销毁栈

bool stackFind(Stack\*&, int); //查找栈中是否有id为第二个参数的元素，有则返回1，否则0

**五、算法描述**

【数据结构】

汽车：一个整型carId存储汽车牌照，一个长整型inTime存储进入时间。

typedef struct {

int carId;

long long inTime;

}Car;

队列：队列queue按要求模拟候车场。

typedef struct {

Car data[MaxSize];

int top;

}Stack;

栈：栈stack按要求模拟停车场，另一个栈temp用做出车时的临时存储。

typedef struct {

Car data[MaxSize];

int top;

}Stack;

【设计思路】

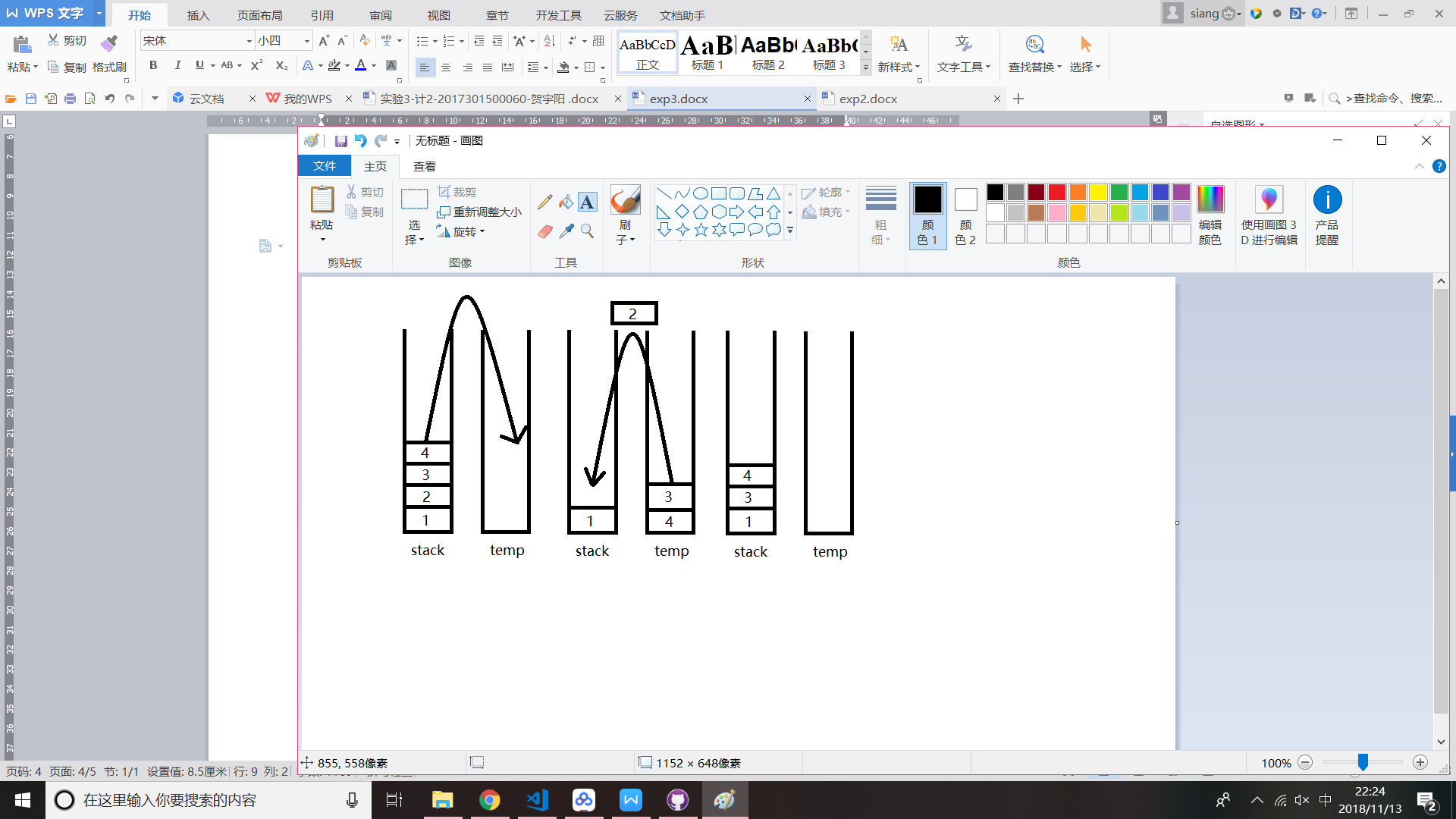
（1）停车：如果停车场未满则直接放入栈，否则进候车场队列排队。当前时间用ctime头文件中的time(NULL)获取绝对秒数。

时间复杂度O(1)

if (getLenth(stack) < n) push(stack, formatCar(id, time(NULL)));

else inQueue(queue, formatCar(id, time(NULL)));

1. 取车：如果停车场有该车牌号的车，则把该车顶上所有的车按出栈stack顺序压入栈temp，取车计算价钱后，将栈temp中的汽车重新压回停车场栈stack。如果栈stack有空位且候车队列queue有车等待则进入汽车，停车时间应该从这里算起。



取出编号为2的汽车

时间复杂度O(n)

printf("Please enter the license plate number:");

scanf("%d", &id);

if (!stackFind(stack, id)) printf("Not found 404.");

else {

while (getTop(stack).carId != id) 出栈stack进栈temp

long long outTime = time(NULL);

Car now = getTop(stack);

付款 CostPerSecond \* (outTime - now.inTime));

pop(stack);

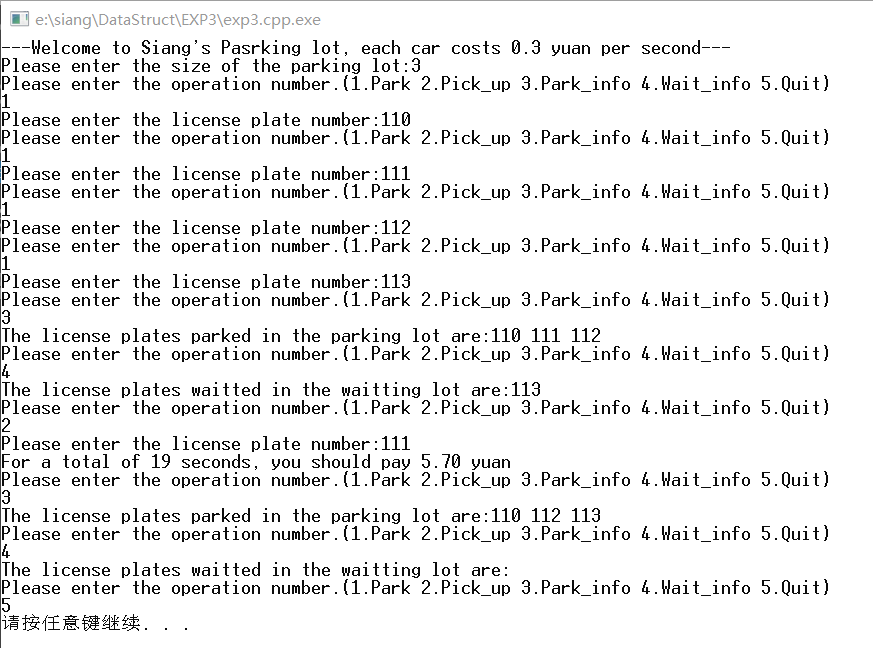
while (!stackEmpty(temp)) 出栈temp进栈stack

while (getLenth(stack) < n && !queueEmpty(queue)) 出队列queue进栈stack

}

**六、实验数据和实验结果分析**

运行结果良好。



**七、实验体会**

停车场问题只不过是为了熟悉栈和队列的综合使用而强行凑的题目，对目的而言还是达到了，就是跟现实停车场相去甚远，问题的有趣性也不及前两个实验，总体而言中规中矩地完成了，不过每个操作为了省事没有单独设置一个函数。这个问题我为了方便计算价钱，定价是0.3元每秒，都是读取在操作时候的系统时间，可能有点黑心哦。