

# 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

二〇一八 年 11 月

## 一、实验题目

### 实验三：停车场管理程序

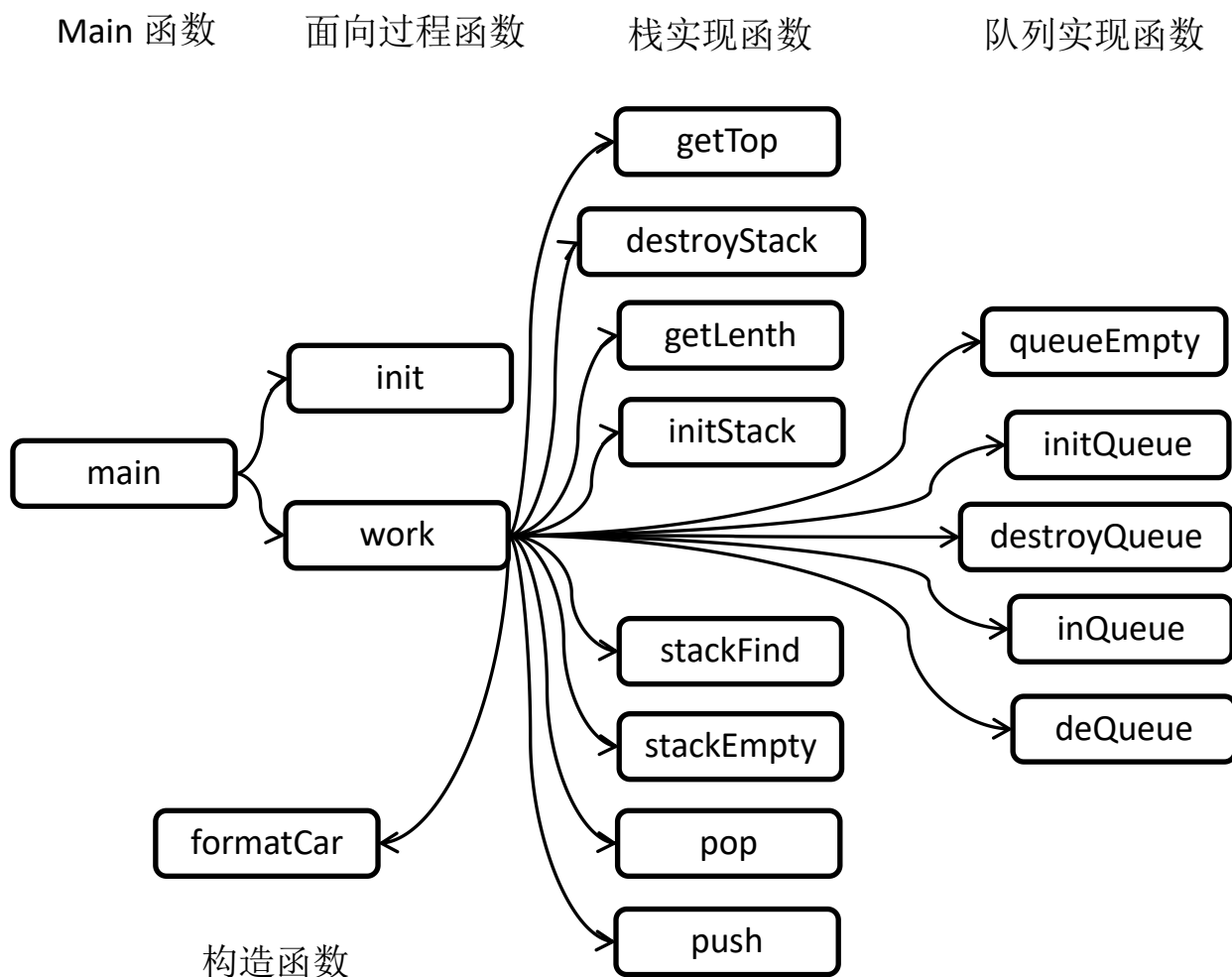
#### 【问题描述】

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。

## 三、实验项目程序结构



## 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 为不为空
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;
```

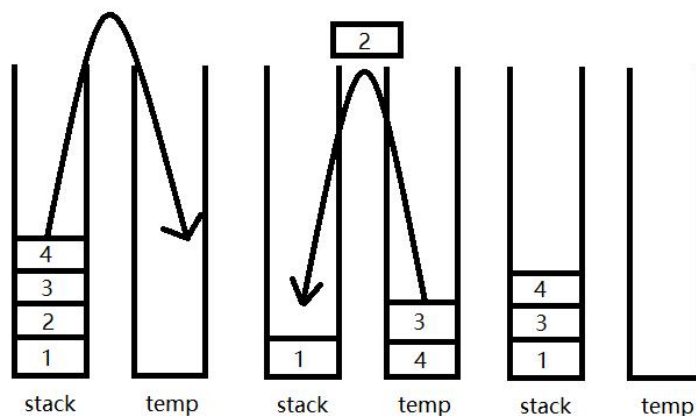
### 【设计思路】

(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)));
```

(2) 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。

```
---Welcome to Siang's Pasrking lot, each car costs 0.3 yuan per second---
Please enter the size of the parking lot:3
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
1
Please enter the license plate number:110
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
1
Please enter the license plate number:111
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
1
Please enter the license plate number:112
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
1
Please enter the license plate number:113
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
3
The license plates parked in the parking lot are:110 111 112
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
4
The license plates waitted in the waitting lot are:113
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
2
Please enter the license plate number:111
For a total of 19 seconds, you should pay 5.70 yuan
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
3
The license plates parked in the parking lot are:110 112 113
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
4
The license plates waitted in the waitting lot are:
Please enter the operation number.(1.Park 2.Pick_up 3.Park_info 4.Wait_info 5.Quit)
5
请按任意键继续. . .
```

## 七、实验体会

停车场问题只不过是為了熟悉棧和隊列的綜合使用而強行湊的題目，對目的而言還是達到了，就是跟現實停車場相去甚遠，問題的有趣性也不及前兩個實驗，總體而言中規中矩地完成了，不過每個操作為了省事沒有單獨設置一個函數。這個問題我為了方便計算價錢，定價是 0.3 元每秒，都是讀取在操作時候的系統時間，可能有點黑心哦。