

数 据 结 构 课 程 设 计

任 务 书

东北大学软件学院

2016 年 9 月

一. 课程基本信息.....	3
二. 教学目的和基本要求:	3
三. 课程设计的内容和安排.....	3
四. 课程设计的题目.....	4
五. 考核的方法和成绩的评定.....	4
六. 时间的安排.....	4
七. 课程设计报告的格式.....	4
八. 课程设计结果的提交.....	4
九. 课程设计题目说明.....	5

一 课程基本信息

- 1、 课程设计名称：数据结构课程设计
- 2、 课程设计编号：
- 3、 课程设计学分：2 学分
- 4、 课程设计周数（学时）：3 周（32 学时）
- 5、 课程设计授课单位：软件工程系
- 6、 指导方式：集体辅导和个别辅导相结合
- 7、 适用专业：软件工程，软英，软日
- 8、 课程设计教材及主要参考资料：

[1] 《数据结构(用面向对象方法与 c++描述)》，殷人昆、陶永雷等著，清华大学出版社，2004.2

二 教学目的和基本要求：

- 1、 了解并掌握数据结构和算法设计的方法，具备初步的独立分析和设计能力
- 2、 初步掌握软件开发过程的问题分析、系统设计、程序编码和测试等基本方法和技能。
- 3、 提高综合运用所学的理论知识和方法，独立分析和解决问题的能力。
- 4、 训练用系统的观点和软件开发一般规范进行软件的开发，培养软件工作者所具备的科学工作方法和作风。

三. 课程设计的内容和安排

- 1、 问题分析和任务定义：根据设计题目的要求，充分地分析和理解问题，明确问题要求做什么，限制的条件是什么。
- 2、 逻辑设计：对问题中涉及到的操作对象定义相应的数据类型，并按照以数据结构为中心的原则划分模块，定义主程序和各抽象数据类型，逻辑设计的结果应写出每个抽象数据类型的定义（包括数据结构的描述和每个基本操作的功能说明），各个主要模块的算法，并画出模块之间的调用关系图。
- 3、 物理设计：定义相应的存储结构并写出个函数的伪码算法。在这个过程中，要综合考虑系统功能，使得系统结构清晰、合理、简单和易于调试，抽象数据类型的实现尽可能做到数据封装，基本操作的规格说明尽可能明确具体。详细设计的结果是对数据结构和基本操作做出进一步的求精，写出数据结构存储结构的类型定义，写出函数形式的算法框架。
- 4、 程序编码：把详细设计的结果进一步求精为程序设计语言。同时加入一些注解和断言，使程序中逻辑清晰。
- 5、 程序调试和测试：采用自底向上，分模块进行，即先调试低层函数。能够熟练掌握调试工具的各种功能，设计测试数据确定疑点，通过修改程序来证实它或绕过它。调试正确后，认真整理源程序和注释，形成格式和风格良好的源程序清单和结果。
- 6、 结果分析：程序运行结果包括正确的输入及其输出结果、含有错误的输入和输出结果、算法时间复杂度和空间复杂度分析。
- 7、 撰写课程设计报告：总结和提升上述过程和步骤，写出结构严谨、表述清楚、符合设计规范的报告。

四 课程设计的题目

本课程实践完成景区信息管理系统实现，该系统有三个信息部分构成，分别为景点导游线路信息，制订景区道路铺设信息，停车场进出车辆信息。

五 考核的方法和成绩的评定

考核成绩分为两个部分：

- (1) 验收程序部分：占 50%。
按照以下几个指标进行验收
程序的正确性
程序的运行和存储效率
程序代码编写规范程度，代码的可读性、
界面是否友好
程序设计的创新性
- (2) 实验报告（50%）
实验报告结构是否清晰，问题叙述是否具有逻辑性
关键代码设计是否正确无误
程序设计的时间、空间复杂性分析是否正确

六 时间的安排

共三周，由浅入深，每个同学可以根据自身实际情况掌握进度，最后一天验收程序。也可以按照阶段分步进行验收。

七 课程设计报告的格式

- (1) 问题的概述、分析和研究意义
- (2) 数据结构的逻辑设计和物理存储设计
- (3) 重要算法的设计、流程描述和代码描述
- (4) 数据结构的时间、空间复杂性分析和重要算法的复杂度分析。
- (5) 程序的最终实现结果（包括重点界面的抓取，能说明问题的重要实验结果数据的打印或其可视化结果）。
- (6) 参考文献
- (7) 附录：关键数据结构的定义及关键算法的源代码。

八 课程设计结果的提交

- (1) 课程设计报告（一份、A4 纸打印，同时包括一份电子文档）
- (2) 完整的程序系统（电子方式提交），能够对输入产生相应的输出，同时尽量完成可视化演示。
- (3) 源程序文档（电子方式提交），源程序代码要求适当缩进编排、结构清晰、可读性好，对源程序的类说明（如采用面向对象方法设计）、函数说明、接口说明、关键变量说明等要进行注释。
- (4) 答辩报告（制作 PPT 报告，电子方式提交），要求重点突出，思路清晰，同时就此报告准备答辩。
所有电子方式提交的文件全部存放在一个目录中，并对其进行压缩，压缩后的文件按规定格式进行命名，命名格式为：班级号+学号+姓名。

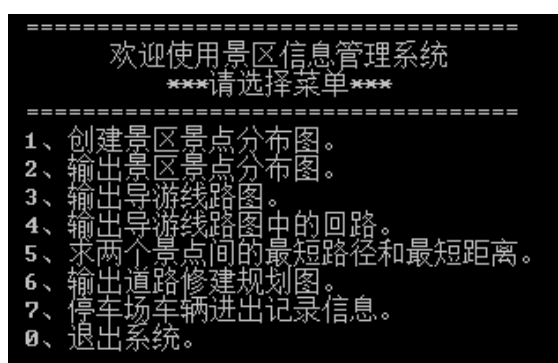
九 课程设计题目说明

1. 建立主程序应用菜单选项

描述

主程序应用菜单选项包含：创建景区景点分布图，输出景区景点分布图，输出导游线路图，判断导游线路图有无回路，两个景点间的最短路径和最短距离，停车场车辆进出记录信息，输出道路修建规划图。

对选项采用数字标识进行选择，对其他错误输入可以进行判别，提示输入错误。

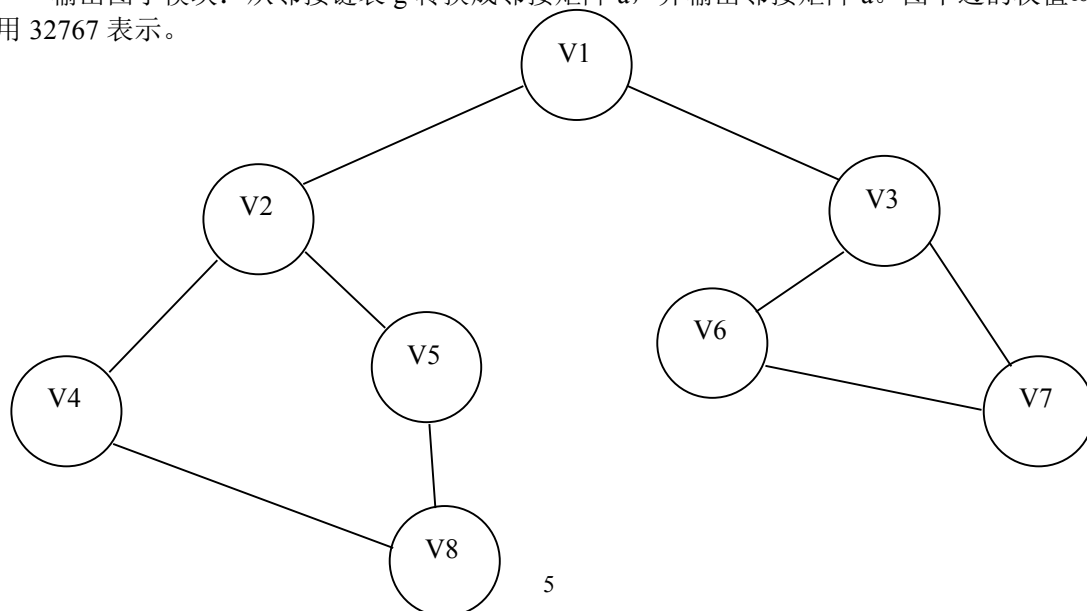


2. 导游线路图的创建

景点的信息包括景点的名称和近邻景点之间的通路和距离。用邻接链表存储景点分布图的信息，（带权无向）图的邻接链表。首要完成创建景区景点分布图和输出景区景点分布图（邻接矩阵）

建图子模块建立无向带权图，输入顶点信息和边的信息，输出邻接链表 G 。由于是无向边，输入一条边时构建两条边。定点信息至少包括：景点名称，景点简介，景点欢迎度，有无休息区，有无公厕。边的信息可以包括两个景点的距离和需要的时间。

输出图子模块：从邻接链表 g 转换成邻接矩阵 a ，并输出邻接矩阵 a 。图中边的权值 ∞ 用 32767 表示。



遍历结点序列与导游线路图转换的策略：

设遍历结果为 $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_i \rightarrow v_{i+1} \rightarrow \dots \rightarrow v_n$

对于结点 v_i 和 v_{i+1} ，如果 v_i 和 v_{i+1} 存在边，则直接转换。

否则，加入边 $v_i \rightarrow v_{i-1}$ ，如果 v_{i-1} 和 v_{i+1} 存在边，则加入边 $v_{i-1} \rightarrow v_{i+1}$ 。

再否则，加入边 $v_{i-1} \rightarrow v_{i-2}$ ，如果 v_{i-2} 和 v_{i+1} 存在边，则加入边 $v_{i-2} \rightarrow v_{i+1}$ 。

如果 v_{i-2} 和 v_{i+1} 还不存在边，继续回溯，一定能找到某个整数 k （因为景点分布图是连通图），使得 v_{i-k} 和 v_{i+1} 存在边，则加入边 $v_{i-k} \rightarrow v_{i+1}$ 。在本任务中，转换后的线路图存于数组 `vex1` 中。

数据的结构体

```
typedef struct ArcNode  
{
```

```
}ArcNode;//定义顶点信息
```

```
typedef struct VNode  
{
```

```
}VNode,AdjList[MAX_VERTEX_NUM];//定义边信息
```

```
typedef struct  
{
```

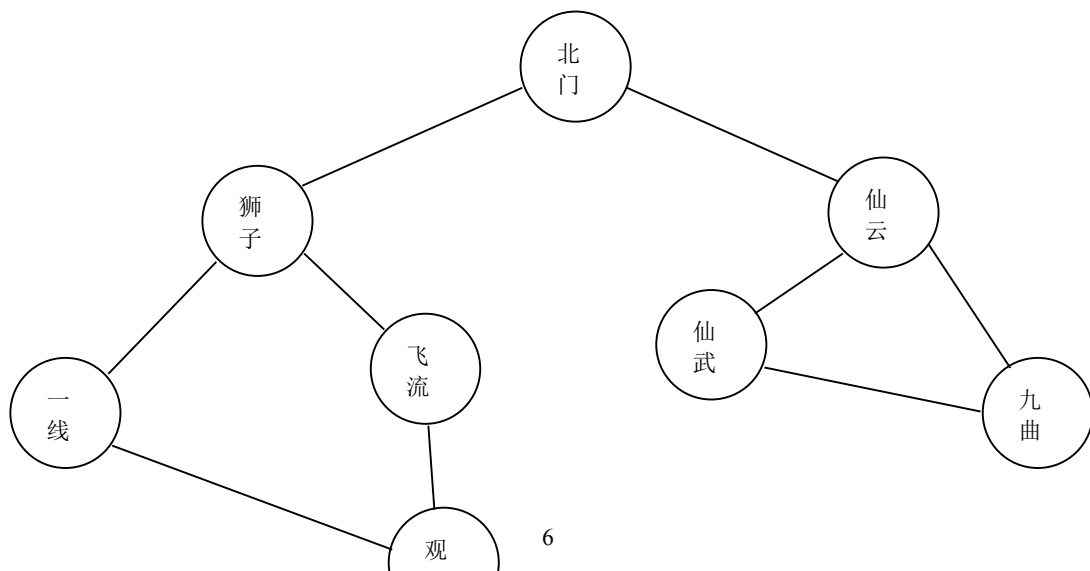
```
}ALGraph;//定义邻接表
```

功能函数：

```
void CreatGraph(ALGraph &G);//创建图的邻接表存储
```

```
void OutputGraph(ALGraph G);//输出图的邻接表
```

对于图的数据信息建立，可以进行多组数据的测试，以保证该函数的功能正确，例如：



```

请输入您要选择的菜单项: 1
请输入顶点数和边数: 8 9

      "请输入各顶点的信息"

请输入各顶点的名字: 北门 狮子山 仙云石 一线天 飞流瀑 仙武湖 九曲桥 观云台
请输入第1条边的两个顶点以及该边的权值: 北门 仙云石 8
请输入第2条边的两个顶点以及该边的权值: 北门 狮子山 9
请输入第3条边的两个顶点以及该边的权值: 狮子山 飞流瀑 6
请输入第4条边的两个顶点以及该边的权值: 狮子山 一线天 7
请输入第5条边的两个顶点以及该边的权值: 一线天 观云台 11
请输入第6条边的两个顶点以及该边的权值: 飞流瀑 观云台 3
请输入第7条边的两个顶点以及该边的权值: 仙云石 九曲桥 5
请输入第8条边的两个顶点以及该边的权值: 仙云石 仙武湖 4
请输入第9条边的两个顶点以及该边的权值: 仙武湖 九曲桥 7

```

```

请输入您要选择的菜单项: 2

```

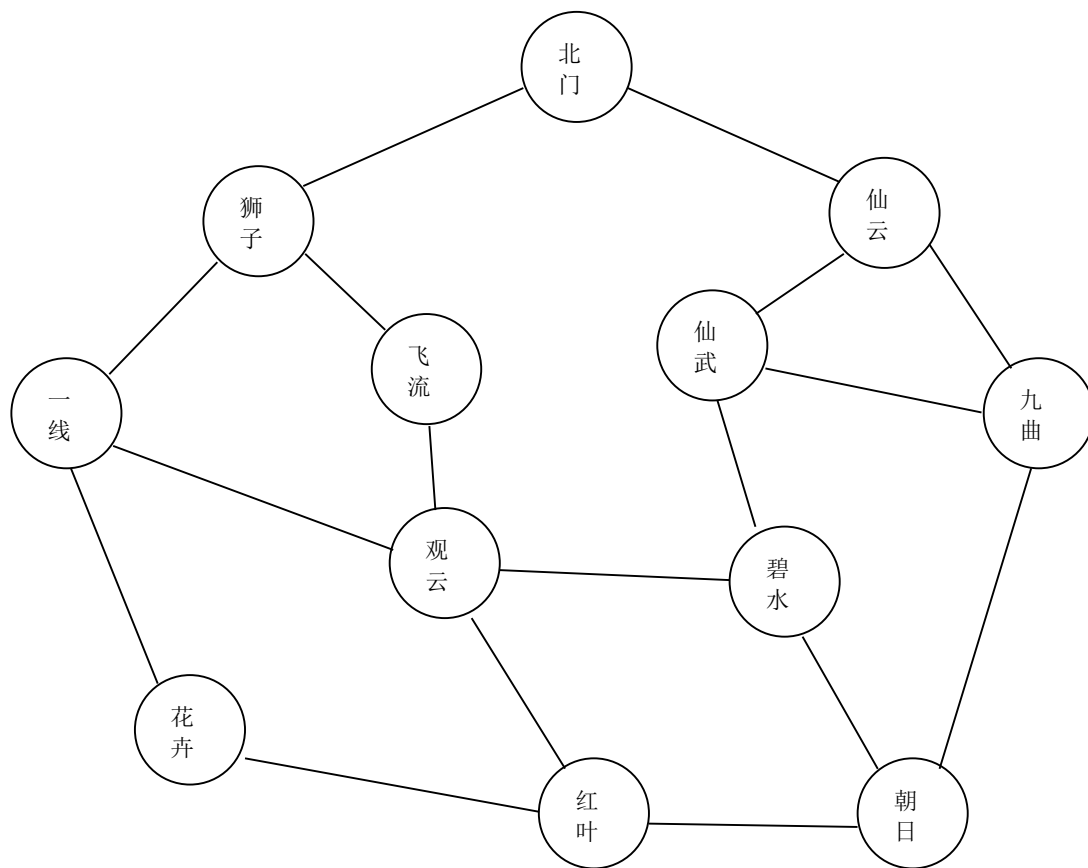
	北门	狮子山	仙云石	一线天	飞流瀑	仙武湖	九曲桥	观云台
北门	0	9	8	32767	32767	32767	32767	32767
狮子山	9	0	32767	7	6	32767	32767	32767
仙云石	8	32767	0	32767	32767	4	5	32767
一线天	32767	7	32767	0	32767	32767	32767	11
飞流瀑	32767	6	32767	32767	0	32767	32767	3
仙武湖	32767	32767	4	32767	32767	0	7	32767
九曲桥	32767	32767	5	32767	32767	7	0	32767
观云台	32767	32767	32767	11	3	32767	32767	0

要求按如下图和权值来进行数据分析
 景点——景点——距离权值

```

北门——狮子山——9
北门——仙云石——8
狮子山——一线天——7
狮子山——飞流瀑——6
仙云石——仙武湖——4
仙云石——九曲桥——5
仙武湖——九曲桥——7
一线天——观云台——11
飞流瀑——观云台——3
一线天——花卉园——10
观云台——红叶亭——15
花卉园——红叶亭——9
观云台——碧水亭——16
仙武湖——碧水亭——20
朝日峰——碧水潭——17
朝日峰——红叶亭——10
九曲桥——朝日峰——20

```



2. 输出导游线路图及其图中的回路部分

景区旅游信息管理系统中制订旅游景点导游线路策略，首先通过遍历景点，给出一个入口景点，建立一个导游线路图，导游线路图用有向图表示。遍历采用深度优先策略，这也比较符合游客心理。

功能函数

输出导游线路图，由 `CreatTourSortGraph(G,G1)` 函数实现；

判断导游线路图有无回路，由 `TopoSort(G1)` 函数实现；

建立路线图

图的深度遍历函数：`void DFSTraverse(ALGraph G);`

判断要查的这两个顶点之间是否有直接相连的边：

`bool IsEdge(ALGraph G,string v1,string v2);`

寻找要查找顶点位置：`int LocateVex(ALGraph G,string v);`

例如：

```
请输入您要选择的菜单项：3
导游路线为：北门狮子山一线天观云台飞流瀑观云台一线天狮子山北门仙云石仙武湖九曲桥
```

回路（选做）

为了使导游线路图能够优化，可通过拓扑排序判断图中有无回路，若有回路，则打印输出回路中的景点，供人工优化。

计算每个顶点的入度，存储在 indegree 数组中将入度减为0的顶点入栈，实现函数：

```
void FindInDegree(ALGraph G,int indegree[]);
```

例如：

```
请输入您要选择的菜单项：4
图中有回路，回路为：
狮子山一线天飞流瀑观云台
```

3. 输出两个景点之间最短路径和最短距离

求两个景点间的最短路径和最短距离，由 MiniDistanse(G1,path,D)函数实现；

输出道路修建规划图，在程序中由 MiniSpanTree(G,G.adjlist[0].name)函数实现。

在导游线路图中，还为一些不愿按线路走的游客提供信息服务，比如从一个景点到另一个景点的最短路径和最短距离。在本线路图中将输出任意景点间的最短路径和最短距离。算法采用迪杰斯特拉算法（必做），弗洛伊德算法（选做）。

计算最短路径：void ShortestPath(ALGraph G,int path[][MAX_VERTEX_NUM],
double D[][MAX_VERTEX_NUM]);

求最短路径：void OutPutShortestPath(ALGraph G,int path[][MAX_VERTEX_NUM],
double D[][MAX_VERTEX_NUM],int i,int j);

例如：

```
请输入您要选择的菜单项：5
请输入要查询距离的两个景点的名称：仙云石 飞流瀑
最短路径为：仙云石--北门
北门--狮子山
狮子山--飞流瀑
最短距离为：23
```

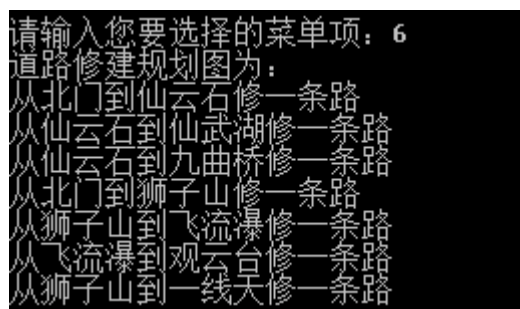
4. 输出道路修建规划图（选做）

在景区建设中，道路建设是其中一个重要内容。道路建设首先要保证能连通所有景点，但又要花最小的代价，可以通过求最小生成树来解决这个问题。本任务中假设修建道路的代价只与它的里程相关。

最小生成树函数：`void MiniSpanTree(ALGraph G,string u);`

实现流程：使用循环初始化各个边的信息，初始化 `u` 已经在最小生成树中可以创建一个函数，将顶点加入生成树中，当新加入的顶点使距离更小了，就更改他。

例如



```
请输入您要选择的菜单项: 6
道路修建规划图为:
从北门到仙云石修一条路
从仙云石到仙武湖修一条路
从仙云石到九曲桥修一条路
从北门到狮子山修一条路
从狮子山到飞流瀑修一条路
从飞流瀑到观云台修一条路
从狮子山到一线天修一条路
```

5、查找及排序

● 查找功能

可以根据用户输入的关键字进行景点的查找，关键字可以在景点名称也可以在景点介绍中。查找成功则返回景点的相关简介，如果查找不成功请给予正确提示。

● 排序功能

可以根据不同的需求进行排序，如按景点欢迎度（必做）或者景点的岔路数对景点进行排序，排序算法可以采用冒泡、快速、插入排序等。

排序后可以直接打印到控制台，检验排序或者查询后的数据数据是否完整。

6. 输出车辆的进出信息

设停车场是一个可以停放 n 辆汽车的南北方向的狭长通道，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，依次由北向南排列（大门在最南端，最先到达的第一辆车停放在车场的最北端），若车场内已停满 n 辆车，那么后来的车只能在门外的便道上等候，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车场为它让路，待该辆车开出大门外，其它车辆再按原次序进入车场，每辆停放在车场的车在它离开停车场时必须按它停留的时间长短交纳费用。试为停车场编制按上述要求进行管理的模拟程序。要求程序输出每辆车到达后的停车位置（停车场或便道上），以及某辆车离开停车场时应缴纳的费用和它在停车场内停留的时间。

设计思路

停车场的管理流程如下：

- ① 当车辆要进入停车场时，检查停车场是否已满，如果未满则车辆进入停车场；如果停车场已满，则车辆进入便道等候。
- ② 当车辆要求出栈时，先让在它之后进入停车场的车辆退出停车场为它让路，再让该车退出停车场，让路的所有车辆再按其原来进入停车场的次序进入停车场。之后，再检查在便道上是否有车等候，有车则让最先等待的那辆车进入停车场。

数据结构

由于停车场只有一个大门，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车场为它让路，先进停车场的后退出，后进车场的先退出，符合栈的“后进先出，先进后出”的操作特点，因此，可以用一个栈来模拟停车场。而当停车场满后，继续来到的其它车辆只能停在便道上，根据便道停车的特点，先排队的车辆先离开便道进入停车场，符合队列的“先进先出，后进后出”的操作特点，因此，可以用一个队列来模拟便道。排在停车场中间的车辆可以提出离开停车场，并且停车场内要离开的车辆之后到达的车辆都必须先离开停车场为它让路，然后这些车辆依原来到达停车场的次序进入停车场，因此在前面已设的一个栈和一个队列的基础上，还需要有一个地方保存为了让路离开停车场的车辆，由于先退出停车场的后进入停车场，所以很显然保存让路车辆的场地也应该用一个栈来模拟。因此，本题求解过程中需用到两个栈和一个队列。栈以顺序结构实现，队列以链表结构实现。

车辆出入清单

程序提示：以栈模拟停车场，以队列模拟车场外的便道，按照从终端读入的输入数据序列进行模拟管理。每一组输入数据包括三个数据项：汽车“到达”或“离去”信息、汽车牌照号码以及到达或离去的时刻。对每一组输入数据进行操作后的输出信息为：若是车辆到达，则输出汽车在停车场内或便道上的停车位置；若是车辆离去，则输出汽车在停车场内停留的时间和应交纳的费用（在便道上停留的时间不收费）。

数据结构：

模拟停车场的堆栈的性质；

```
typedef struct zanlind{
    int number; //汽车车号
    int ar_time; //汽车到达时间
}zanlNode;
typedef struct{
    zanlNode *base; //停车场的堆栈底
    zanlNode *top; //停车场的堆栈顶
    int stacksize_current;
}stackhead;
```

例如：

请输入您要选择的菜单项：？

*** 停车场管理程序 ***

```
=====
***                                     ***
***  A --- 汽车 进 车场    D --- 汽车 出 车场    ***
***                                     ***
***          E --- 退出  程序          ***
=====
```

请选择 :<A,D,E>: A

车牌为: 1234

进场的时刻:13:00

该车已进入停车场在: 1 号车道