

# 目录

1. 需求分析.....	1
2. 项目亮点.....	1
3. 概要设计.....	1
4. 详细设计.....	3
4.1 定义.....	3
4.2 管理员模式.....	3
4.3 用户模式.....	4
5. 管理员手册 .....	7
5.1 管理员模式.....	7
5.2 添加城市.....	8
5.3 删除城市.....	9
5.4 编辑交通.....	9
5.5 查询路线.....	10
6. 用户手册.....	10
7. 心得体会.....	12
8. 附录 .....	13
8.1 definition.h .....	13
8.2 city.c .....	14
8.3 admin.c .....	16
8.4 function.c.....	20
8.5 main.c .....	22
8.6 baidu_map_api.py.....	23
8.7 tmp.html .....	25

## 1. 需求分析

- (1) 建立全国城市的公路网（包含高速公路、普通道路及其城市之间所需相应的时间和费用），编制汽车自驾出行的交通咨询程序，为旅客提供两种最优决策的交通咨询。
- (2) 提供对城市信息进行编辑（如添加、删除）的功能。
- (3) 提供两种最优决策：最快到达或最省钱到达。
- (4) 咨询以用户和计算机的对话方式进行。由用户输入起点、终点，最优决策原则，输出信息：最快需要多长时间或最少需要多少费用才能到达，并列出具体的中间经过的城市和道路。

## 2. 项目亮点

- (1) 利用百度地图 API 爬取了 392 个省市的名字和经纬度，以及 6 万条两个城市驾车路线的时间和费用，具有现实参考意义。
- (2) 建立了密码登录（不显示输入）的管理员模式，实现管理员与用户的独立使用。
- (3) 用户在查询后可以实时显示出路线图，该图是基于百度地图 API 绘制的。
- (4) 建立了深色设置，用户可以根据现实时间自主调节白天或夜间模式，管理员的密码登录口隐藏在深色设置中。
- (5) 管理员可以在系统或文件中查看并修改相关信息，并实时保存。
- (6) 用户在选择查询后可以选择查看另一种方案。
- (7) 程序分割为三个库，使得项目的调用更加清晰合理。

## 3. 概要设计

首先读取数据 `city.txt` 和 `city.csv`，建立城市信息结构体及无向图的数据结构。

然后选择白天/夜间模式，管理员可以使用隐藏命令进入密码登录口，输入密码进入管理员模式，退出管理员模式时会自动回到用户模式。

用户在查询时，使用弗洛伊德算法，在用户输入起始城市和目标城市并选择了决策原则后，生成相应的最短路径。

整体的功能使用和输入合法性检验运用了 C 语言的 `goto` 函数，直到输入满足相应的选择后会执行其他程序。

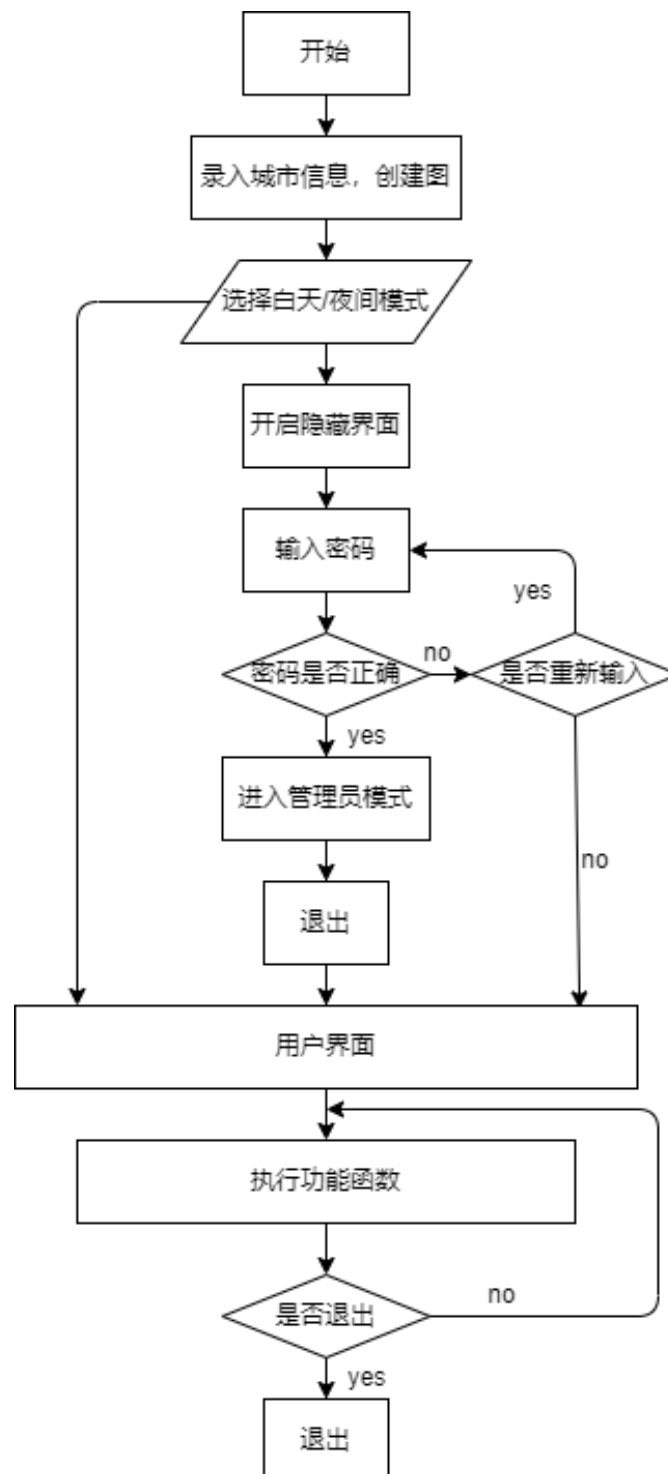


图 1: 主函数流程图

## 4. 详细设计

### 4.1 定义

运用两个邻接矩阵分别存储交通路线的时间和费用，将程序分割为 city.c、admin.c 和 function.c 进行相关程序的编写。

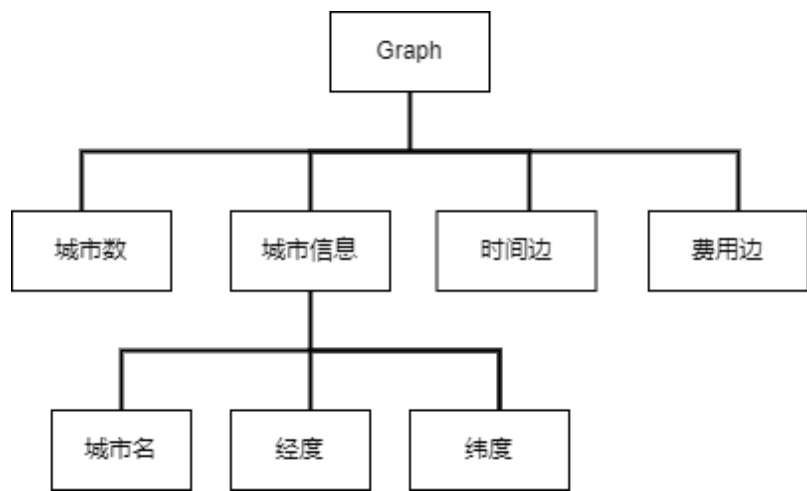


图 2: 数据结构示意图

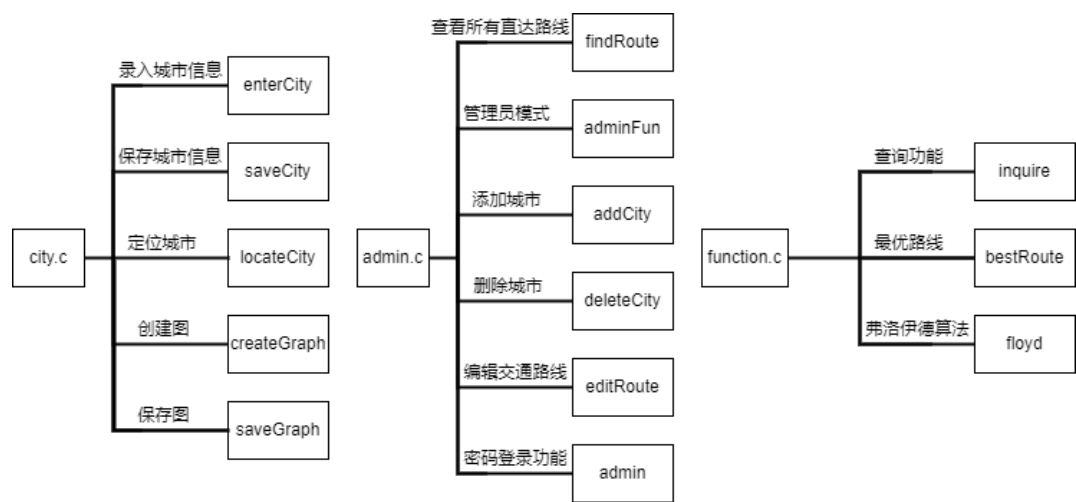


图 3: 函数结构示意图

### 4.2 管理员模式

在退出管理员模式前，保存城市信息和路线信息到 city.txt 和 city.csv 文件中。

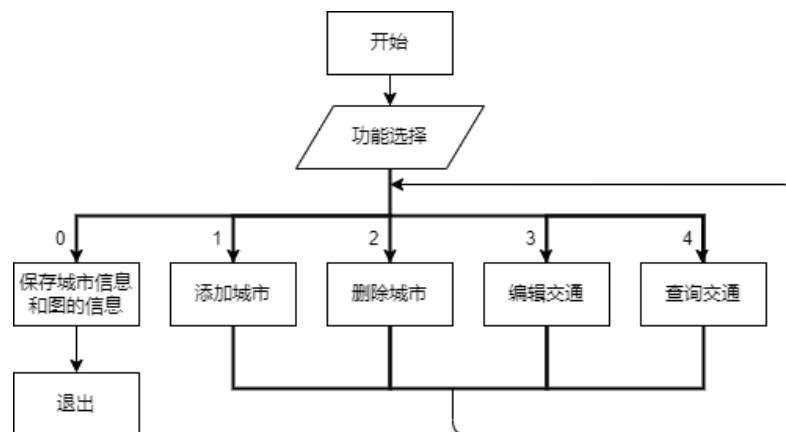


图 4: 管理员模式流程图

### 4.3 用户模式

关于路线图的绘制，首先将 tmp.html 复制为新的以“起始城市-目标城市”为文件名的 html 文件，然后通过 C 语言进行编写，最后使用 C 语言 shellapi 库的函数在浏览器打开文件。

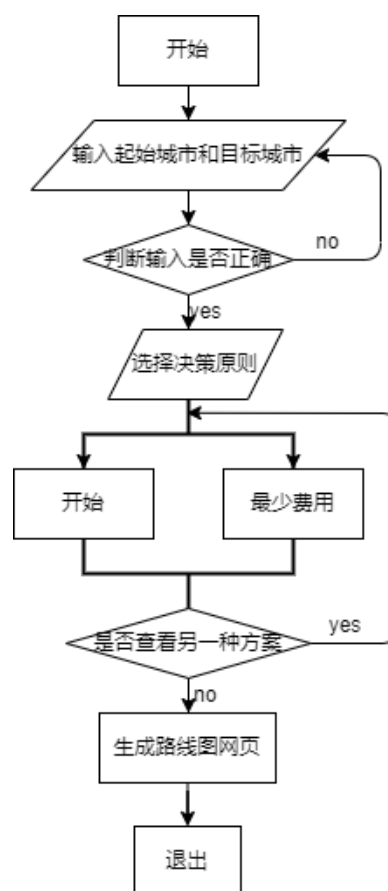


图 5: 用户模式流程图

```

1 Status inquire(Graph *G) //查询路线
2 {
3     char name1[20],name2[20],choice,name[100],filename[105],order[120];;
4     int i,j,city1,city2,choose,flag;
5     printf("请输入起始城市名称:");
6     fflush(stdin);scanf("%s",name1);
7     city1=locateCity(G,name1);
8     if(city1<0) printf("%s城市不存在!\n",name1);
9     else
10    {
11        printf("请输入目标城市名称:");
12        fflush(stdin);scanf("%s",name2);
13        city2=locateCity(G,name2);
14        if(city2<0) printf("%s城市不存在!\n",name2);
15        else if(city1==city2) printf("请勿重复输入城市!\n");
16        else
17        {
18            printf("=====\n");
19            printf("1. 最快到达\n2. 最少费用\n");
20            printf("=====\n");
21            A:printf("请选择决策原则:");
22            scanf("%d",&choose);
23            if(choose!=1&&choose!=2) {printf("输入错误,请重新输入!\n");goto A;}
24            flag=bestRoute(G,city1,city2,choose);
25            printf("\n是否查看另一种方案(y|n):");
26            fflush(stdin);scanf("%c",&choice);
27            if(choice=='y')
28            {
29                choose=3-choose;
30                flag=bestRoute(G,city1,city2,choose);
31            }
32            if(flag)
33            {
34                strcpy(order,"copy tmp.html ");strcpy(name,G->city[city1].name);
35                strcat(name,"——");strcat(name,G->city[city2].name);
36                strcpy(filename,name);
37                strcat(filename,".html");
38                strcat(order,filename);strcat(order,">nul");
39                system(order);
40                fp=fopen(filename,"a");
41                fprintf(fp,"\nvar start=new
42                    BMap.Point(%lf,%lf);\n",G->city[city1].lng,G->city[city1].lat);
43                fprintf(fp,"var end=new
44                    BMap.Point(%lf,%lf);\n",G->city[city2].lng,G->city[city2].lat);
45                fprintf(fp,"driving.search(start,end);\n</script>\n</body>\n</html>");
46                fclose(fp);

```

```

45         system("pause");
46         ShellExecute(NULL, "open", filename, NULL, NULL, SW_SHOWNORMAL);
47     }
48 }
49 }
50 }
51
52 Status bestRoute(Graph *G, int city1, int city2, int flag) //最优路线
53 {
54     int sum=0;
55     if(flag==1)
56         floyd(G->citynum, G->timeEdges);
57     else
58         floyd(G->citynum, G->costEdges);
59     if(weight[city1][city2] != INF)
60     {
61         if(flag==1)
62             printf("\n%s--->%s的最少用时为: %.2lfh\t", G->city[city1].name,
63                 G->city[city2].name, weight[city1][city2]/3600.0);
64         else printf("\n%s--->%s的最少花费为: %d $ \t", G->city[city1].name,
65                 G->city[city2].name, weight[city1][city2]);
66         int x=path[city1][city2], a=b, result=0;
67         a=city1;
68         printf("路径为: %s", G->city[city1].name);
69         while(x!=-1)
70         {
71             printf("-->%s", G->city[x].name);
72             if(flag==1) result+=G->costEdges[a][x];
73             else result+=G->timeEdges[a][x];
74             a=x;
75             x=path[city1][x];
76         }
77         printf("-->%s\t", G->city[city2].name);
78         if(flag==1) {result+=G->costEdges[a][city2]; printf("花费为 %d $ \n", result);}
79         else {result+=G->timeEdges[a][city2]; printf("用时为 %.2lfh\n", result/3600.0);}
80         return 1;
81     }
82     else printf("%s和%s之间不可达!\n", G->city[city1].name, G->city[city2].name);
83     return 0;
84 }
85
86 Status floyd(int n, int edge[][MAX]) //弗洛伊德算法
87 {
88     int i, j, k;
89     for(i=0; i<n; i++)
90     {
91         for(j=0; j<n; j++)

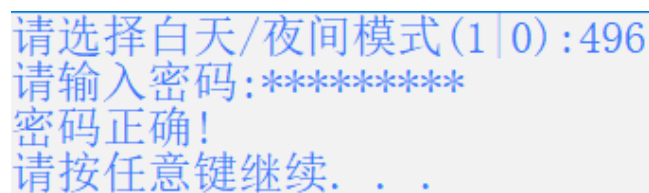
```

```
92     {
93         weight[i][j]=edge[i][j];
94         path[i][j]=-1;
95     }
96 }
97 for(k=0;k<n;k++)
98 {
99     for(i=0;i<n;i++)
100     {
101         for(j=0;j<n;j++)
102         {
103             if(weight[i][j]>weight[i][k]+weight[k][j])
104             {
105                 weight[i][j]=weight[i][k]+weight[k][j];
106                 path[i][j]=k;
107             }
108         }
109     }
110 }
111 }
```

## 5. 管理员手册

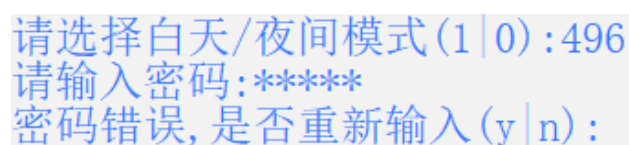
管理员账号：33550336，密码：admin8128，登录口：496，数字都是完全数。

### 5.1 管理员模式



请选择白天/夜间模式(1|0):496  
请输入密码:\*\*\*\*\*  
密码正确!  
请按任意键继续. . .

图 6: 密码登录口



请选择白天/夜间模式(1|0):496  
请输入密码:\*\*\*\*\*  
密码错误, 是否重新输入(y|n):

图 7: 密码输入错误时



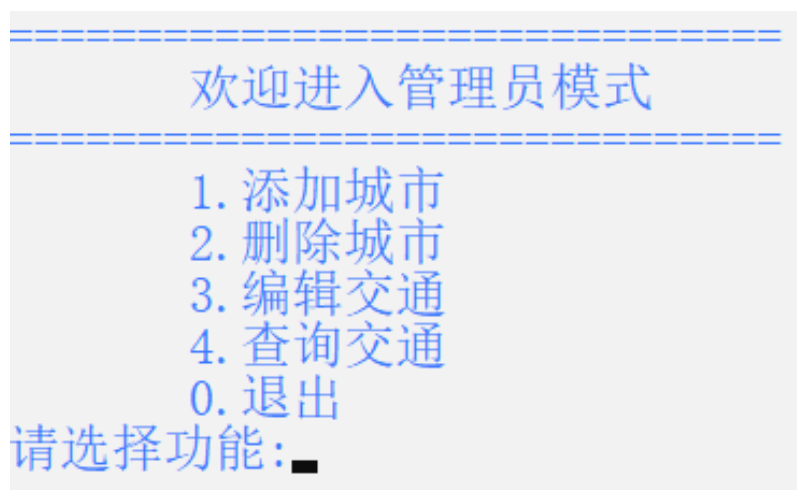


图 8: 管理员模式界面

## 5.2 添加城市

```
请输入城市名称:台北
请输入台北的经度:121.520076
请输入台北的纬度:25.030724
添加成功!
是否继续添加城市(y|n):
```

图 9: 添加城市

```
请输入城市名称:上海
上海城市已存在!
是否继续添加城市(y|n):
```

图 10: 添加城市重复时

### 5.3 删除城市

```
请输入城市名称:台北
请输入管理员账号确认删除台北城市及其所有路线:33550336
删除成功!
是否继续删除城市(y|n):
```

图 11: 删除城市

```
请输入起始城市名称:州杭
州杭城市不存在!
是否退出(y|n):
```

图 12: 城市不存在时

### 5.4 编辑交通

```
请输入起始城市名称:杭州
请输入目标城市名称:香港
暂无杭州-->香港的路线!
请输入时间(h):15.2
请输入费用(¥):566
编辑成功!
是否继续编辑路线(y|n):
```

图 13: 无路线时编辑交通

```
请输入起始城市名称:杭州
请输入目标城市名称:上海
杭州-->上海      用时:3.13h      花费:71¥
请输入时间(h):0
输入错误, 请重新输入!
请输入时间(h):
```

图 14: 有路线时编辑交通

5.5 查询路线

管理员可以查询某个城市的所有直达路线。

请输入城市名称:杭州		
目标城市	时间	费用
北京	15.74h	598¥
天津	15.19h	400¥
上海	3.13h	71¥
重庆	21.64h	866¥
合肥	5.82h	177¥
芜湖	3.90h	126¥
蚌埠	7.03h	165¥
淮南	6.93h	198¥
马鞍山	4.38h	104¥
淮北	8.05h	213¥
铜陵	4.20h	146¥
安庆	5.18h	212¥
黄山	3.71h	90¥
滁州	5.30h	115¥
阜阳	8.17h	249¥
宿州	7.50h	204¥
六安	7.10h	219¥

图 15: 查询路线部分结果

6. 用户手册

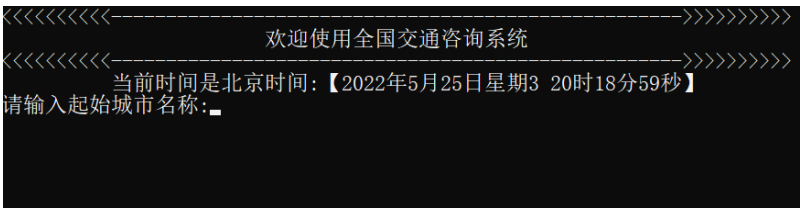


图 16: 夜间模式

```
请输入起始城市名称:杭州
请输入目标城市名称:杭州
请勿重复输入城市!
是否退出(y|n):
```

图 17: 重复输入时

```
<<<<<<<<<<                                     >>>>>>>>>>
      欢迎使用全国交通咨询系统
<<<<<<<<<------>>>>>>>>>>

    当前时间是北京时间：【2022年5月25日星期三 20时50分29秒】
请输入起始城市名称:重庆
请输入目标城市名称:大连
=====
1. 最快到达
2. 最少费用
=====
请选择决策原则:1

重庆—>大连的最少用时为:28.98h 路径为:重庆—>烟台—>大连          花费为919¥

是否查看另一种方案(y|n):y

重庆—>大连的最少花费为:829¥   路径为:重庆—>安康—>大连        用时为32.28h
请按任意键继续. . .
```

图 18: 用户查询结果



图 19: 路线图 (会自动在浏览器打开)

>>>>>>感谢您的使用,下次再会!<<<<<<<<  
请按任意键继续. . .

图 20: 退出系统

## 7. 心得体会

这次课程设计的心得体会通过实践我们的收获如下：

1. 这次的交通咨询模拟课程设计，我们沿用了航空客运订票系统的文件存储函数和时间函数、家庭族谱的功能界面以及最小生成树问题的无向图数据结构，使得程序的编写方便快捷。
2. 结合了之前课程设计的经验，提高了程序的鲁棒性。
3. 学习了 html 文件的编写、Python 基于 API 爬虫的原理和 C 语言 shellapi 库的应用。
4. 这个项目是设计结构课程设计的最后一个项目，它结合了之前八次课程设计的 Debug 思想和编写范式，很好地体现了之前课程设计的收获。

## 8. 附录

### 8.1 definition.h

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4  #include<time.h>
5  #include<malloc.h>
6  #include<conio.h>
7  #include<windows.h>
8  #include<shellapi.h>
9  #define MAX 700
10 #define INF 0x3f3f3f3f
11 //枚举类型返回值
12 typedef int Status;
13 //城市信息
14 typedef struct
15 {
16     double lng;    //经度
17     double lat;    //纬度
18     char name[20]; //城市名
19 }City;
20 //图
21 typedef struct
22 {
23     int citynum;        //城市数
24     City city[MAX];     //城市信息
25     int timeEdges[MAX][MAX]; //时间边(秒)
26     int costEdges[MAX][MAX]; //费用边(元)
27 }Graph;
28 //city.c
29 Status enterCity(Graph *G); //录入城市信息
30 Status saveCity(Graph *G); //保存城市信息
31 Status locateCity(Graph *G, char name[]); //定位城市
32 Status createGraph(Graph *G); //创建图
33 Status saveGraph(Graph *G); //保存图
34 //admin.c
35 Status admin(Graph *G); //密码登录功能
36 Status adminFun(Graph *G); //管理员模式
37 Status addCity(Graph *G); //添加城市
38 Status deleteCity(Graph *G); //删除城市
39 Status editRoute(Graph *G); //编辑交通路线
40 Status findRoute(Graph *G); //查看城市的所有直达路线
41 //function.c
42 Status inquire(Graph *G); //查询功能
```

```
43 Status bestRoute(Graph *G,int city1,int city2,int flag);//最优路线
44 Status floyd(int n,int cost[][MAX]);//弗洛伊德算法
45
46 FILE *fp; //文件指针
47 int weight[MAX][MAX]={0},path[MAX][MAX]={0};//权值矩阵和路径矩阵，便于实现算法
```

## 8.2 city.c

```
1 Status enterCity(Graph *G) //录入城市信息
2 {
3     int i=0;
4     if((fp=fopen("city.txt","r"))==NULL)
5     {
6         printf("该文件夹下无city.txt!\n");
7         system("pause");exit(0);
8     }
9     while(fscanf(fp,"%s %lf %lf",G->city[i].name,&G->city[i].lng,&G->city[i].lat)!=EOF)
10         i++;
11     fclose(fp);
12     G->citynum=i;
13 }
14 Status saveCity(Graph *G) //保存城市信息
15 {
16     fp=fopen("city.txt","w");
17     for(int i=0;i<G->citynum;i++)
18         fprintf(fp,"%s %lf %lf\n",G->city[i].name,G->city[i].lng,G->city[i].lat);
19     fclose(fp);
20 }
21
22 Status locateCity(Graph *G,char name[]) //定位城市
23 {
24     for(int i=0;i<G->citynum;i++)
25     {
26         if(strcmp(G->city[i].name,name)==0)
27             return i;
28     }
29     return -1;
30 }
31
32 Status createGraph(Graph *G) //创建图
33 {
34     char *line,*record,buffer[1024],name1[20],name2[20];
35     int i,j,city1,city2;
36     if((fp=fopen("city.csv","r"))==NULL)
```

```

37     {
38         printf("该文件夹下无city.csv!\n");
39         system("pause");exit(0);
40     }
41     for(i=0;i<G->citynum;i++)
42     {
43         for(j=0;j<G->citynum;j++)
44         {
45             if(i==j)
46             {
47                 G->timeEdges[i][j]=0;
48                 G->costEdges[i][j]=0;
49             }
50             else
51             {
52                 G->timeEdges[i][j]=INF;
53                 G->costEdges[i][j]=INF;
54             }
55         }
56     }
57     fgets(buffer,sizeof(buffer),fp);
58     while((line=fgets(buffer,sizeof(buffer),fp)))
59     {
60         record=strtok(line,",");
61         if(record!=NULL)
62         {
63             sprintf(name1,"%s",record);record=strtok(NULL, ",");
64             sprintf(name2,"%s",record);record=strtok(NULL, ",");
65             city1=locateCity(G,name1);city2=locateCity(G,name2);
66             G->timeEdges[city1][city2]=atoi(record);record=strtok(NULL, ",");
67             G->costEdges[city1][city2]=atoi(record);record=strtok(NULL, ",");
68             G->timeEdges[city2][city1]=G->timeEdges[city1][city2];
69             G->costEdges[city2][city1]=G->costEdges[city1][city2];
70         }
71     }
72     fclose(fp);
73 }
74
75 Status saveGraph(Graph *G) //保存图
76 {
77     int i,j;
78     fp=fopen("city.csv","w");
79     fprintf(fp,"起始城市,目标城市,时间,费用\n");
80     for(i=0;i<G->citynum;i++)
81     {
82         for(j=i+1;j<G->citynum;j++)
83         {

```



```
84         if(G->timeEdges[i][j]!=INF&&G->costEdges[i][j]!=INF)
85             fprintf(fp,"%s,%s,%d,%d\n",G->city[i].name,G->city[j].name,
86                 G->timeEdges[i][j],G->costEdges[i][j]);
87     }
88 }
89 fclose(fp);
90 }
```

### 8.3 admin.c

```
1  Status admin(Graph *G) //密码登录功能
2  {
3      int i;
4      char password[20],choice;
5      A:printf("请输入密码:");fflush(stdin);
6      while(i<19&&(password[i]=getch())!='\r')
7      {
8          if(password[i]=='\b')
9          {
10             printf("\b b");
11             i--;
12         }
13         else
14         {
15             putchar('*');
16             i++;
17         }
18     }
19     password[i]='\0';
20     if(strcmp(password,"admin8128")==0)
21     {
22         printf("\n密码正确!\n");
23         system("pause");adminFun(G);
24     }
25     else
26     {
27         printf("\n密码错误,是否重新输入(y|n):");
28         fflush(stdin);scanf("%c",&choice);
29         if(choice=='y') goto A;
30     }
31     saveCity(G);saveGraph(G);
32 }
33
34 Status adminFun(Graph *G) //管理员模式
35 {
```

```

36     int choose;
37     A:system("cls");
38     printf("=====\n");
39     printf("    欢迎进入管理员模式\n");
40     printf("=====\n");
41     printf("    1.添加城市\n");
42     printf("    2.删除城市\n");
43     printf("    3.编辑交通\n");
44     printf("    4.查询交通\n");
45     printf("    0.退出\n");
46     printf("请选择功能:");scanf("%d",&choose);
47     switch(choose)
48     {
49         case 1:addCity(G);goto A;
50         case 2:deleteCity(G);goto A;
51         case 3:editRoute(G);goto A;
52         case 4:findRoute(G);goto A;
53         default:break;
54     }
55 }
56
57 Status addCity(Graph *G) //添加城市
58 {
59     char name[20],choice;
60     A:system("cls");
61     printf("请输入城市名称:");
62     fflush(stdin);scanf("%s",name);
63     if(locateCity(G,name)>=0) printf("%s城市已存在!\n",name);
64     else
65     {
66         printf("请输入%s的经度:",name);scanf("%lf",&G->city[G->citynum].lng);
67         printf("请输入%s的纬度:",name);scanf("%lf",&G->city[G->citynum].lat);
68         strcpy(G->city[G->citynum].name,name);
69         G->citynum++;
70         printf("添加成功!\n");
71     }
72     printf("是否继续添加城市(y|n):");
73     fflush(stdin);scanf("%c",&choice);
74     if(choice=='y') goto A;
75 }
76
77 Status deleteCity(Graph *G) //删除城市
78 {
79     char name[20],choice;
80     int i,delete,del;
81     A:system("cls");
82     printf("请输入城市名称:");

```

```

83     fflush(stdin);scanf("%s",name);
84     del=locateCity(G,name);
85     if(del<0) printf("%s城市不存在!\n",name);
86     else
87     {
88         B:printf("请输入管理员账号确认删除%s城市及其所有路线:",name);
89         scanf("%d",&delete);
90         if(delete!=33550336)
91         {
92             printf("账号错误,是否重新输入(y|n):");
93             fflush(stdin);scanf("%c",&choice);
94             if(choice=='y') goto B;
95         }
96         else
97         {
98             for(i=0;i<G->citynum;i++)
99             {
100                 G->timeEdges[del][i]=INF;
101                 G->costEdges[del][i]=INF;
102                 G->timeEdges[i][del]=INF;
103                 G->costEdges[i][del]=INF;
104             }
105             printf("删除成功!\n");
106         }
107     }
108     printf("是否继续删除城市(y|n):");
109     fflush(stdin);scanf("%c",&choice);
110     if(choice=='y') goto A;
111 }
112
113 Status editRoute(Graph *G) //编辑交通路线
114 {
115     char name1[20],name2[20],choice;
116     int i,j,city1,city2,cost;
117     double time;
118     A:system("cls");
119     printf("请输入起始城市名称:");
120     fflush(stdin);scanf("%s",name1);
121     city1=locateCity(G,name1);
122     if(city1<0) printf("%s城市不存在!\n",name1);
123     else
124     {
125         printf("请输入目标城市名称:");
126         fflush(stdin);scanf("%s",name2);
127         city2=locateCity(G,name2);
128         if(city2<0) printf("%s城市不存在!\n",name2);
129         else

```

```

130     {
131         if(G->timeEdges[city1][city2]!=INF&&G->costEdges[city1][city2]!=INF)
132             printf("%s--->%s\t用时:%.2lfh\t花费:%d$ \n",name1,name2,
133                 G->timeEdges[city1][city2]/3600.0,G->costEdges[city1][city2]);
134         else printf("暂无%s-->%s的路线!\n",name1,name2);
135         B:printf("请输入时间(h):");scanf("%lf",&time);
136         time*=3600;
137         if(time<=0||time>=INF) {printf("输入错误,请重新输入!\n");goto B;}
138         G->timeEdges[city1][city2]=(int)time;
139         G->timeEdges[city2][city1]=(int)time;
140         C:printf("请输入费用($):");scanf("%d",&cost);
141         if(cost<=0||cost>=INF) {printf("输入错误,请重新输入!\n");goto C;}
142         G->costEdges[city1][city2]=cost;
143         G->costEdges[city2][city1]=cost;
144         printf("编辑成功!\n");
145     }
146 }
147 printf("是否继续编辑路线(y|n):");
148 fflush(stdin);scanf("%c",&choice);
149 if(choice=='y') goto A;
150 }
151
152 Status findRoute(Graph *G) //查看城市的所有路线
153 {
154     char name[20],choice;
155     int i,find;
156     A:system("cls");
157     printf("请输入城市名称:");
158     fflush(stdin);scanf("%s",name);
159     find=locateCity(G,name);
160     if(find<0) printf("%s城市不存在!\n",name);
161     else
162     {
163         printf("目标城市   时间   费用\n");
164         for(i=0;i<G->citynum;i++)
165         {
166             if(i!=find&&G->timeEdges[find][i]!=INF&&G->costEdges[find][i]!=INF)
167                 printf("%s\t%.2lfh\t%d$ \n",G->city[i].name,
168                     G->timeEdges[find][i]/3600.0,G->costEdges[find][i]);
169         }
170     }
171     printf("是否继续查看交通(y|n):");
172     fflush(stdin);scanf("%c",&choice);
173     if(choice=='y') goto A;
174 }

```

## 8.4 function.c

```

1 Status inquire(Graph *G) //查询路线
2 {
3     char name1[20],name2[20],choice,name[100],filename[105],order[120];;
4     int i,j,city1,city2,choose,flag;
5     printf("请输入起始城市名称:");
6     fflush(stdin);scanf("%s",name1);
7     city1=locateCity(G,name1);
8     if(city1<0) printf("%s城市不存在!\n",name1);
9     else
10    {
11        printf("请输入目标城市名称:");
12        fflush(stdin);scanf("%s",name2);
13        city2=locateCity(G,name2);
14        if(city2<0) printf("%s城市不存在!\n",name2);
15        else if(city1==city2) printf("请勿重复输入城市!\n");
16        else
17        {
18            printf("=====\n");
19            printf("1.最快到达\n2.最少费用\n");
20            printf("=====\n");
21            A:printf("请选择决策原则:");
22            scanf("%d",&choose);
23            if(choose!=1&&choose!=2) {printf("输入错误,请重新输入!\n");goto A;}
24            flag=bestRoute(G,city1,city2,choose);
25            printf("\n是否查看另一种方案(y|n):");
26            fflush(stdin);scanf("%c",&choice);
27            if(choice=='y')
28            {
29                choose=3-choose;
30                flag=bestRoute(G,city1,city2,choose);
31            }
32            if(flag)
33            {
34                strcpy(order,"copy tmp.html ");strcpy(name,G->city[city1].name);
35                strcat(name,"——");strcat(name,G->city[city2].name);
36                strcpy(filename,name);
37                strcat(filename,".html");
38                strcat(order,filename);strcat(order,">nul");
39                system(order);
40                fp=fopen(filename,"a");
41                fprintf(fp,"\nvar start=new
                        BMap.Point(%lf,%lf);\n",G->city[city1].lng,G->city[city1].lat);
42                fprintf(fp,"var end=new
                        BMap.Point(%lf,%lf);\n",G->city[city2].lng,G->city[city2].lat);

```

```

43         fprintf(fp,"driving.search(start,end);\n</script>\n</body>\n</html>");
44         fclose(fp);
45         system("pause");
46         ShellExecute(NULL,"open",filename,NULL,NULL,SW_SHOWNORMAL);
47     }
48 }
49 }
50 }
51
52 Status bestRoute(Graph *G,int city1,int city2,int flag) //最优路线
53 {
54     int sum=0;
55     if(flag==1)
56         floyd(G->citynum,G->timeEdges);
57     else
58         floyd(G->citynum,G->costEdges);
59     if(weight[city1][city2]!=INF)
60     {
61         if(flag==1)
62             printf("\n%s--->%s的最少用时为:%.2lfh\t",G->city[city1].name,
63                 G->city[city2].name,weight[city1][city2]/3600.0);
64         else printf("\n%s--->%s的最少花费为:%d $\t",G->city[city1].name,
65                 G->city[city2].name,weight[city1][city2]);
66         int x=path[city1][city2],a,b,result=0;
67         a=city1;
68         printf("路径为:%s",G->city[city1].name);
69         while(x!=-1)
70         {
71             printf("-->%s",G->city[x].name);
72             if(flag==1) result+=G->costEdges[a][x];
73             else result+=G->timeEdges[a][x];
74             a=x;
75             x=path[city1][x];
76         }
77         printf("-->%s\t",G->city[city2].name);
78         if(flag==1) {result+=G->costEdges[a][city2];printf("花费为%d $ \n",result);}
79         else {result+=G->timeEdges[a][city2];printf("用时为%.2lfh\n",result/3600.0);}
80         return 1;
81     }
82     else printf("%s和%s之间不可达!\n",G->city[city1].name,G->city[city2].name);
83     return 0;
84 }
85
86 Status floyd(int n,int edge[][MAX]) //弗洛伊德算法
87 {
88     int i,j,k;
89     for(i=0;i<n;i++)

```

```
90     {
91         for(j=0;j<n;j++)
92         {
93             weight[i][j]=edge[i][j];
94             path[i][j]=-1;
95         }
96     }
97     for(k=0;k<n;k++)
98     {
99         for(i=0;i<n;i++)
100         {
101             for(j=0;j<n;j++)
102             {
103                 if(weight[i][j]>weight[i][k]+weight[k][j])
104                 {
105                     weight[i][j]=weight[i][k]+weight[k][j];
106                     path[i][j]=k;
107                 }
108             }
109         }
110     }
111 }
```

## 8.5 main.c

```
1  #include "definition.h"
2  #include "city.c"
3  #include "admin.c"
4  #include "function.c"
5  Status main()
6  {
7      system("color F9");
8      int color;
9      char choice;
10     Graph *G;
11     G=(Graph*)malloc(sizeof(Graph));
12     time_t T;
13     time(&T);
14     struct tm *now;
15     now=localtime(&T);
16     enterCity(G);createGraph(G);
17     printf("请选择白天/夜间模式(1|0):");scanf("%d",&color);
18     if(color==496) admin(G);
19     else if(!color) system("color 0F");
20     A:system("cls");
```

```

21     printf("<<<<<<<<<----->>>>>>>>>\n");
22     printf("                欢迎使用全国交通咨询系统\n");
23     printf("<<<<<<<<<----->>>>>>>>>\n");
24     printf("                当前时间是北京时间：【%d年%d月%d日 星期%d\n",now->tm_year+1900,
                %d时%d分%d秒】 \n",now->tm_year+1900,
25         now->tm_mon+1,now->tm_mday,now->tm_wday,now->tm_hour,now->tm_min,now->tm_sec);
26     inquire(G);
27     printf("是否退出(y|n):");fflush(stdin);scanf("%c",&choice);
28     if(choice=='y')
29     {
30         printf("\n>>>>>>>感谢您的使用,下次再见!<<<<<<<<\n");
31         system("pause");exit(0);
32     }
33     system("pause");goto A;
34 }

```

## 8.6 baidu\_map\_api.py

```

1  import os
2  from typing import Optional
3  import sys
4  import pandas as pd
5  import requests
6
7  def read_df(file_path: str) -> pd.DataFrame:
8      """
9      读取csv文件
10     @param file_path: 文件路径
11     @return: 数据框
12     """
13     file_path = os.path.join(os.path.dirname(__file__), file_path)
14     return pd.read_csv(file_path)
15
16  class BaiduMap:
17      """
18      使用该处理所有有关百度地图API的操作
19      """
20      def __init__(self):
21          self.directionlite_url = 'http://api.map.baidu.com/directionlite/v1/driving'
22          self.direction_url = 'http://api.map.baidu.com/direction/v2/driving'
23          self.session = requests.Session()
24
25      def login(self, ak: str) -> None:
26          """
27          使用百度地图API需要提供AK

```



```
28     @param ak: 从百度地图API的控制台处获得的AK
29     """
30     self.ak = ak
31
32     def read_lng_lat(self, file_path: str) -> None:
33         """
34         读取城市经纬度
35         @param file_path: 经纬度数据存储路径
36         """
37         lng_lat_df = read_df(file_path)
38         self.lng_lat = {}
39         for index in range(len(lng_lat_df)):
40             self.lng_lat[lng_lat_df.at[index, 'city']] = (round(lng_lat_df.at[index,
41                 'longitude'], 6), round(lng_lat_df.at[index, 'latitude'], 6))
42
43     def get_lng_lat(self, city_name: str) -> tuple:
44         """
45         获取城市经纬度
46         @param city_name: 城市名字
47         @return: 城市的经纬度
48         """
49         return self.lng_lat[city_name]
50
51     def fetch_route(self, origin: list, destination: list, method: str =
52         'directionlite') -> dict:
53         """
54         使用百度地图API获取所需要的信息(里程数,驾车时间,过路费)
55         @param origin: 起点的经纬度, 经度在前, 纬度在后
56         @param destination: 终点的经纬度, 经度在前, 纬度在后
57         @param method: 使用轻量路线规划或路线规划API,取值为directionlite或direction
58         @return: 返回想要写入数据框中的信息
59         """
60         if method == 'directionlite':
61             url = self.directionlite_url
62         elif method == 'direction':
63             url = self.direction_url
64         params = {
65             'origin': f'{origin[1]},{origin[0]}',
66             'destination': f'{destination[1]},{destination[0]}',
67             'ak': self.ak,
68         }
69         resp = self.session.get(url = url, params = params).json()
70         if resp['status'] == 0:
71             routes = resp['result']['routes'][0]
72             return {
73                 'distance': routes['distance'] / 1000,
74                 'duration': routes['duration'],
```

```

73         'toll': routes['toll'],
74     }
75     else:
76         print(params)
77         raise Exception('调用出错! ')
78
79 if __name__ == "__main__":
80     import time
81     map = BaiduMap()
82     map.login('Your Baidu Map API AK')
83     map.read_lng_lat('cities_lng_lat.csv')
84     result_df = read_df('distance.csv')
85     print('使用百度地图API读取数据...')
86     for index in range(len(result_df)):
87         if result_df.at[index, 'baidu_map_distance'] == 0.0:
88             origin_city = result_df.at[index, 'origin']
89             destination_city = result_df.at[index, 'destination']
90             origin_position = map.get_lng_lat(origin_city)
91             destination_position = map.get_lng_lat(destination_city)
92             try:
93                 resp = map.fetch_route(origin_position, destination_position,
94                                       method='direction')
95             except:
96                 print('运行出错了, 速速存盘保平安! ')
97                 result_df.to_csv(os.path.join(os.path.dirname(__file__), 'distance.csv'),
98                               index=False)
99                 sys.exit()
100             result_df.at[index, 'baidu_map_distance'] = resp['distance']
101             result_df.at[index, 'duration'] = resp['duration']
102             result_df.at[index, 'toll'] = resp['toll']
103             if index % 20 == 19:
104                 print(f'处理到第 {index + 1} 条数据, 准备存盘...')
105                 result_df.to_csv(os.path.join(os.path.dirname(__file__), 'distance.csv'),
106                               index=False)

```

## 8.7 tmp.html

```

1 <!DOCTYPE html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
5 <title>交通咨询模拟</title>
6 <script type="text/javascript"
7     src="https://api.map.baidu.com/api?v=3.0&ak=0Vq04p0L5sAPVCP6zGW94HT9Wrs7tG2x">

```

```
8 <style type="text/css">
9 body, html, #container {width: 100%;height: 100%;overflow:
    hidden;margin:0;font-family:"微软雅黑";}
10 </style>
11 </head>
12 <body>
13 <div id="container"></div>
14 <script type="text/javascript">
15 var map=new BMap.Map("container");
16 map.centerAndZoom(new BMap.Point(116.404, 39.915),14);
17 var driving=new BMap.DrivingRoute(map,{renderOptions:{map:map,autoViewport:true }});
```