

数据结构说明和数据字典

熊宇

1. 首先设计结点结构体

```
typedef struct
{
    double time;
    char loc;
}Node;
```

2. 全局变量

```
double len[NumOfNode + 2];    //表示源点到 i 这个点的距离
int visit[NumOfNode + 2];    //节点是否被访问
int n = NumOfNode + 1; //结点数
Node NodeList[250]; //存储结点的数组
double Mar[NumOfNode + 2][NumOfNode + 2]; //结点的邻接矩阵
double cost[NumOfNode + 2][NumOfNode + 2]; //结点间经过处理后的邻接矩阵
int path[NumOfNode + 2][NumOfNode + 2]; //存储中间结点的数组
string OutLoc[NumOfNode + 2][NumOfNode + 2]; //存储路径的数组
int timepath[NumOfNode + 2][NumOfNode + 2][NumOfNode + 2]; //存储经过的所有结点的数组
```

3. 一些函数

```
void SystemInit(Node * l, double M[][NumOfNode + 2]); //内置模块，初始化函数
void Examine(Node * l, double M[][NumOfNode + 2]); //Debug 时所用的检验函数
double Search(char c); //查找对应城市的对应风险值
void GetIns(double cost[][NumOfNode + 2], Node* l, double M[][NumOfNode + 2], double len[NumOfNode + 2], int visit[NumOfNode + 2], int n, int path[][NumOfNode + 2], string OutLoc[][NumOfNode + 2], int timepath[][NumOfNode + 2][NumOfNode + 2]); //捕捉用户命令进行线路分析及输出
```