# **实验十三 最小生成树Prim算法**

## **问题**

农民约翰被选为他们镇的镇长！他其中一个竞选承诺就是在镇上建立起互联网，并连接到所有的农场。当然，他需要你的帮助。 约翰已经给他的农场安排了一条高速的网络线路，他想把这条线路共享给其他农场。为了用最小的消费，他想铺设最短的光纤去连接所有的农场。 你将得到一份各农场之间连接费用的列表，你必须找出能连接所有农场并所用光纤最短的方案。 每两个农场间的距离不会超过100000。

## **实验思路**

贪心算法，需要从一个点出发，寻找最近的一个点，然后将两个点作为一个整体，再次去寻找最近的一个点。

## **实验步骤**

有两个一维数组和一个二维数组，cost储存权值，lowcost储存最短路径长度，closet储存最短路径。

## **代码**

#**include**<iostream>

#**define** MAX\_VALUE 100000

**using** **namespace** std;

**int** lowcost[1000],cost[100][100],closest[1000];

**int** **prim**(**int** v0,**int** N){

**int** i, j, mindis, minone; **int** ans = 0;/\*用来记录最小生成树的总长度\*/

**for**(i = 0; i < N; i++) {

lowcost[i] = cost[v0][i];

closest[i] = v0;

}

**for**(i = 0; i < N-1; i++) {

mindis = MAX\_VALUE;

**for**(j = 0; j < N; j++)

**if**(lowcost[j] && mindis > lowcost[j]) {

mindis = lowcost[j];

minone = j;

} /\*将找到的最近点加入最小生成树\*/

ans += lowcost[minone];

lowcost[minone] = 0; /\*修正其他点到最小生成树的距离\*/

**for**(j = 0; j < N; j++)

**if**(cost[j][minone] < lowcost[j]) {

lowcost[j] = cost[j][minone];

closest[j] = minone;

}

} **return** ans;

}

**int** **main**(){

**int** N;

**while**(cin>>N){

**for**(**int** i=0;i<N;i++)

**for**(**int** j=0;j<N;j++)

cin>>cost[i][j];

cout<<prim(0,N);

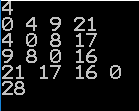
}

}

## **测试**

正常

## **结果**



## **总结**

邻接矩阵的主对角线一定是0，因为是同样的点，这个在输入的时候要引起注意。当有顶点并入最短路径的数组中后，要去修正到其他点的最短距离，并判断。