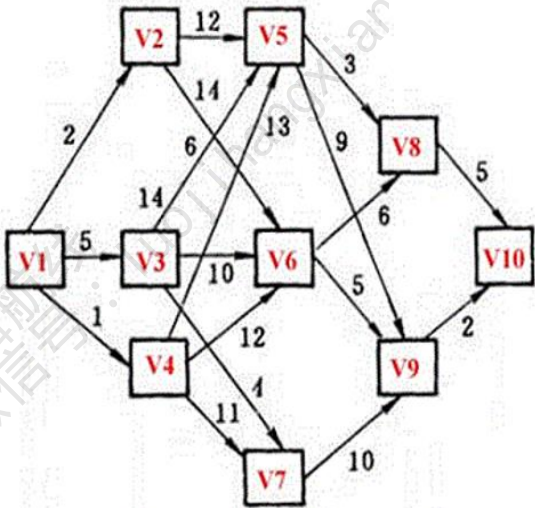


1261：城市交通路网

题目描述

下图表示城市之间的交通路网，线段上的数字表示费用，单向通行由A->E。试用动态规划的最优化原理求出A->E的最省费用。



如图：求v1到v10的最短路径长度及最短路径。

输入

第一行为城市的数量N;
后面是N*N的表示两个城市间费用组成的矩阵。

输出

求v1到v10的最短路径长度及最短路径。

输入样例

```
10
0 2 5 1 0 0 0 0 0 0
0 0 0 0 12 14 0 0 0 0
0 0 0 0 6 10 4 0 0 0
0 0 0 0 13 12 11 0 0 0
0 0 0 0 0 0 0 3 9 0
0 0 0 0 0 0 0 6 5 0
0 0 0 0 0 0 0 0 10 0
0 0 0 0 0 0 0 0 0 5
0 0 0 0 0 0 0 0 0 2
0 0 0 0 0 0 0 0 0 0
```

输出样例

minlong=19
1 3 5 8 10

解析

本题的描述有误，请参考本PDF。

我们将邻接矩阵转化成表格，展示如下：

	1	2	3	4	5	6	7	8	9	10
1	0	2	5	1	0	0	0	0	0	0
2	0	0	0	0	12	14	0	0	0	0
3	0	0	0	0	6	10	4	0	0	0
4	0	0	0	0	13	12	11	0	0	0
5	0	0	0	0	0	0	0	3	9	0
6	0	0	0	0	0	0	0	6	5	0
7	0	0	0	0	0	0	0	0	10	0
8	0	0	0	0	0	0	0	0	0	5
9	0	0	0	0	0	0	0	0	0	2
10	0	0	0	0	0	0	0	0	0	0

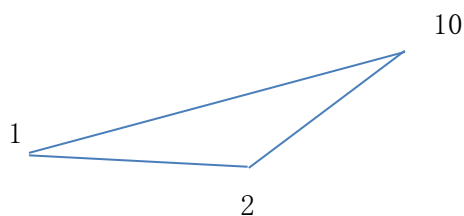
两点(i,x)之间的数字表示它们之间的距离，我们用 $a[i][x]$ 表示。

我们用 $f[i]$ 表示第i个点到最后一个点的最短距离。

则有 $f[n]=0$ ，即最后一个点到自己的距离必然为0。

进一步分析，我们可以得出 $f[i] = \min(a[i][x] + f[x], f[i])$;

如下图所示：



$f[1] = \min(f[1], a[1][2] + f[2])$, 根据公式, 我们发现, 我想知道 $f[1]$ 就必须知道 $f[2]$, 想知道 $f[2]$ 就必须要知道 $f[3]$, 直至最后, 我们知道 $f[n]$ 必然为0。

因此, 我们需要不断的从后向前比较, 直到计算出 $f[1]$ 的值, 就是我们的最终答案。

另外一个问题是求中途路径, 这个很容易, 我们只需要将过程中的关键点存储, 然后打印即可。

```
#include <bits/stdc++.h>

using namespace std;
const int M = 105;
const int MaxDis = 0x3f;
int a[M][M]; //两个城市之间的原始距离
int f[M]; //从第i点到最后一个点的最短距离
int c[M]; //c记录下个位置点

int main() {
    int i, j, n, x;
    cin >> n;
    //读入长度数据
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= n; j++) {
            cin >> a[i][j];
        }
        //到达第i个城市的距离, 我们设置为最大
        f[i] = MaxDis;
    }
    //最后一个点到最后一个点的距离必然为0
    f[n] = 0;
    //从后向前, 找到距离最短的组合
    //以示例为例, 我们需要计算从第9个点到最后一个点的最短距离
    for (i = n - 1; i >= 1; i--) {
        //找到两点之间更短的距离
        //a[i][x] > 0 : 两点之间有距离
        //f[i] > a[i][x] + f[x] : 两点之间的距离更短
        for (x = i + 1; x <= n; x++) {
            if (a[i][x] > 0 && (f[i] > a[i][x] + f[x])) {
                //更新第i个点到最后一个点的最短距离
                f[i] = a[i][x] + f[x];
                //记录最短路径点
                c[i] = x;
            }
        }
    }
    cout << "minlong=" << f[1] << endl;
    x = 1;
    //打印最短路径点
    while (x != 0) {
```

```
    cout << x << " ";  
    x = c[x];  
}  
cout << endl;  
return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

