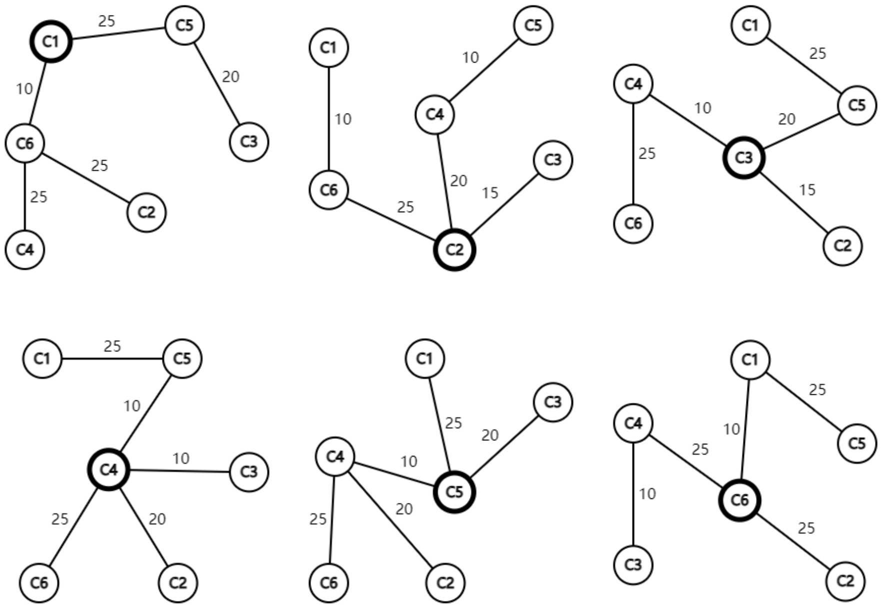
第一次图论作业

姓名：郑有为 学号：19335286

1. 使用 Dijkstra 算法获得每一个城市节点 Ci 关于其它 5 个城市节点的最短路径树。

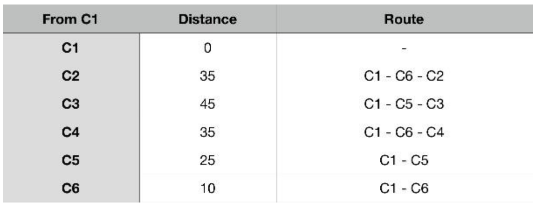
结果：

1. 六棵最短路径树（其中加粗的节点为根节点，最短路径树表示从该节点出发到其余各节点的最短路径）

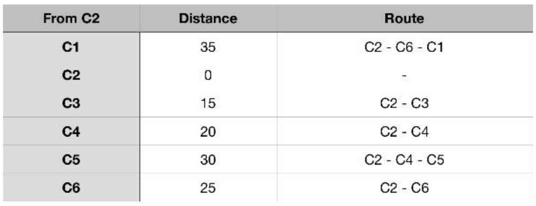


无向图生成网站：https://csacademy.com/app/graph_editor/

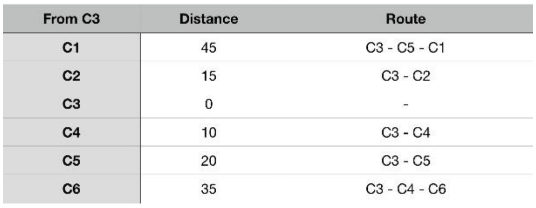
2.最短距离路径表：



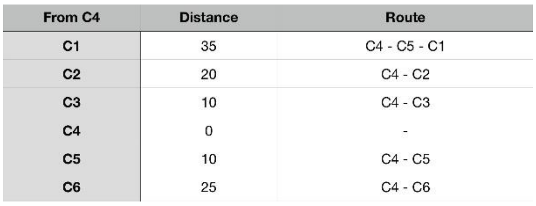
From C1	Distance	Route
C1	0	-
C2	35	C1 - C6 - C2
C3	45	C1 - C5 - C3
C4	35	C1 - C6 - C4
C5	25	C1 - C5
C6	10	C1 - C6



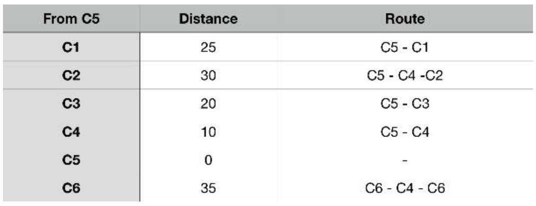
From C2	Distance	Route
C1	35	C2 - C6 - C1
C2	0	-
C3	15	C2 - C3
C4	20	C2 - C4
C5	30	C2 - C4 - C5
C6	25	C2 - C6



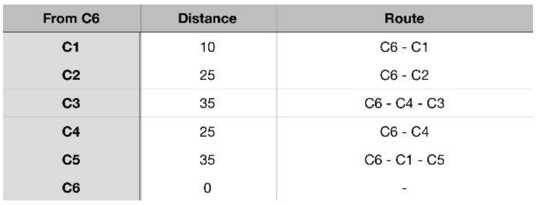
From C3	Distance	Route
C1	45	C3 - C5 - C1
C2	15	C3 - C2
C3	0	-
C4	10	C3 - C4
C5	20	C3 - C5
C6	35	C3 - C4 - C6



From C4	Distance	Route
C1	35	C4 - C5 - C1
C2	20	C4 - C2
C3	10	C4 - C3
C4	0	-
C5	10	C4 - C5
C6	25	C4 - C6



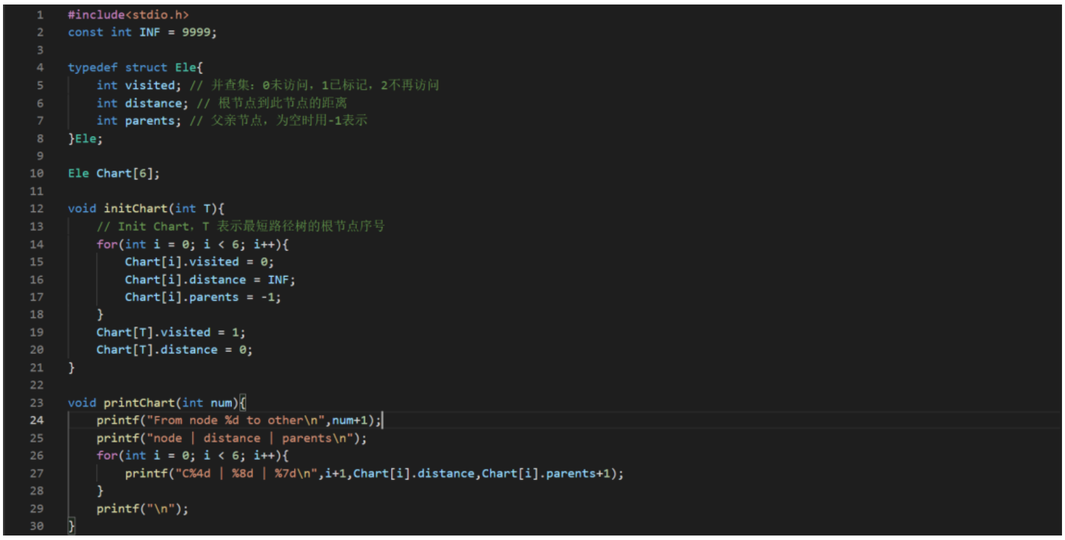
From C5	Distance	Route
C1	25	C5 - C1
C2	30	C5 - C4 -C2
C3	20	C5 - C3
C4	10	C5 - C4
C5	0	-
C6	35	C6 - C4 - C6



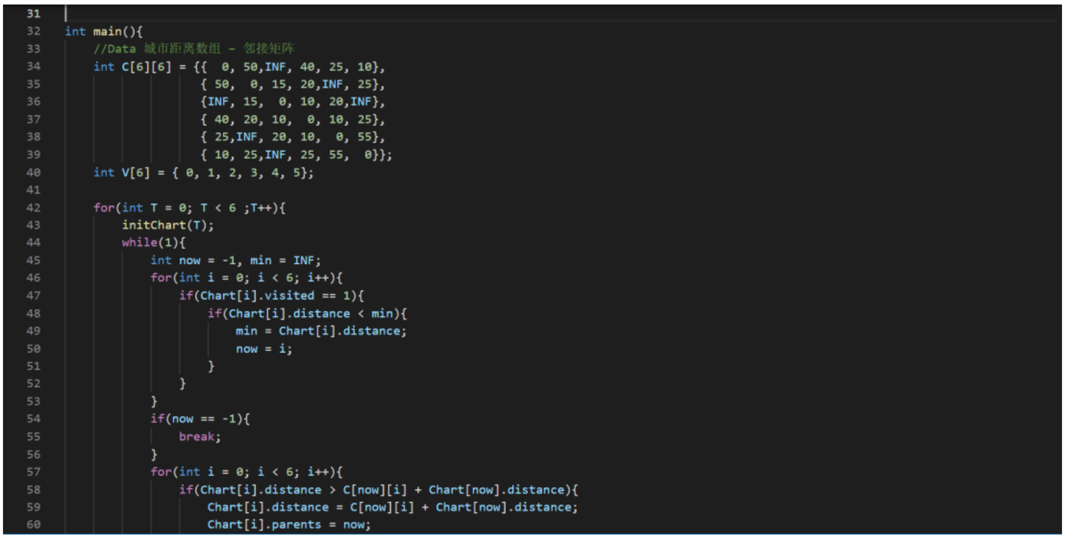
From C6	Distance	Route
C1	10	C6 - C1
C2	25	C6 - C2
C3	35	C6 - C4 - C3
C4	25	C6 - C4
C5	35	C6 - C1 - C5
C6	0	-

附：

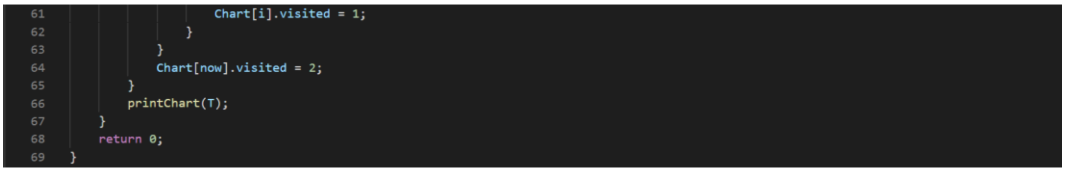
代码（C 语言）：



```c
#include<stdio.h>
const int INF = 9999;

typedef struct Ele{
    int visited; // 并查集：0未访问，1已标记，2不再访问
    int distance; // 根节点到此节点的距离
    int parents; // 父亲节点，为空时用-1表示
}Ele;

Ele Chart[6];

void initChart(int T){
    // Init Chart，T 表示最短路径树的根节点序号
    for(int i = 0; i < 6; i++){
        Chart[i].visited = 0;
        Chart[i].distance = INF;
        Chart[i].parents = -1;
    }
    Chart[T].visited = 1;
    Chart[T].distance = 0;
}

void printChart(int num){
    printf("From node %d to other\n",num+1);
    printf("node | distance | parents\n");
    for(int i = 0; i < 6; i++){
        printf("C%4d | %8d | %7d\n",i+1,Chart[i].distance,Chart[i].parents+1);
    }
    printf("\n");
}
```

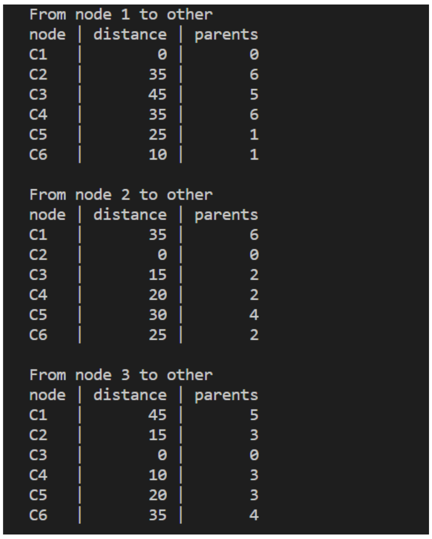


```c

int main(){
    //Data 城市距离数组 - 邻接矩阵
    int C[6][6] = {{  0, 50,INF, 40, 25, 10},
                   { 50,  0, 15, 20,INF, 25},
                   {INF, 15,  0, 10, 20,INF},
                   { 40, 20, 10,  0, 10, 25},
                   { 25,INF, 20, 10,  0, 55},
                   { 10, 25,INF, 25, 55,  0}};
    int V[6] = { 0, 1, 2, 3, 4, 5};

    for(int T = 0; T < 6 ;T++){
        initChart(T);
        while(1){
            int now = -1, min = INF;
            for(int i = 0; i < 6; i++){
                if(Chart[i].visited == 1){
                    if(Chart[i].distance < min){
                        min = Chart[i].distance;
                        now = i;
                    }
                }
            }
            if(now == -1){
                break;
            }
            for(int i = 0; i < 6; i++){
                if(Chart[i].distance > C[now][i] + Chart[now].distance){
                    Chart[i].distance = C[now][i] + Chart[now].distance;
                    Chart[i].parents = now;
```

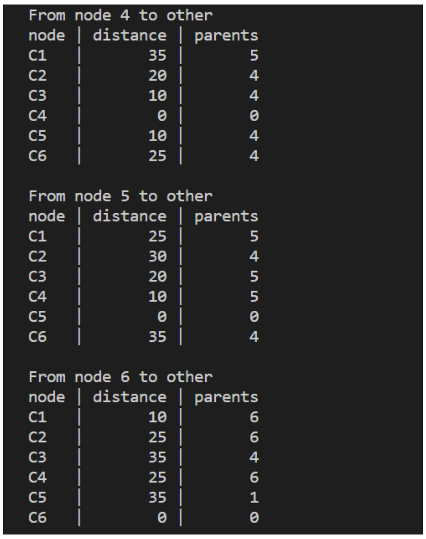


```c
                    Chart[i].visited = 1;
                }
            }
            Chart[now].visited = 2;
        }
        printChart(T);
    }
    return 0;
}
```

程序运行截图：



```
From node 1 to other
node | distance | parents
C1   |        0 |       0
C2   |       35 |       6
C3   |       45 |       5
C4   |       35 |       6
C5   |       25 |       1
C6   |       10 |       1

From node 2 to other
node | distance | parents
C1   |       35 |       6
C2   |        0 |       0
C3   |       15 |       2
C4   |       20 |       2
C5   |       30 |       4
C6   |       25 |       2

From node 3 to other
node | distance | parents
C1   |       45 |       5
C2   |       15 |       3
C3   |        0 |       0
C4   |       10 |       3
C5   |       20 |       3
C6   |       35 |       4
```



```
From node 4 to other
node | distance | parents
C1   |       35 |       5
C2   |       20 |       4
C3   |       10 |       4
C4   |        0 |       0
C5   |       10 |       4
C6   |       25 |       4

From node 5 to other
node | distance | parents
C1   |       25 |       5
C2   |       30 |       4
C3   |       20 |       5
C4   |       10 |       5
C5   |        0 |       0
C6   |       35 |       4

From node 6 to other
node | distance | parents
C1   |       10 |       6
C2   |       25 |       6
C3   |       35 |       4
C4   |       25 |       6
C5   |       35 |       1
C6   |        0 |       0
```