《数据结构》实验报告T008

班 级: DL062138 姓 名: 张苏宇 学 号: 2021302853

E-mail:905159071@qq.com 日 期: 2022/6/4

**实验4.1：**

**◎实验题目:**

**求赋权图中一个节点到所有节点的最短路径的长度**

**◎实验内容：**

**给一个赋权图（无向图），求0号节点到其余所有节点的最短路径的长度。**

1. **需求分析**

**输入**：先输入一个小于等于100的正整数n，然后输入赋权图的邻接矩阵（10000表示无穷大，并且任意一条简单路径的长度都小于10000）

**输出：**按节点编号的顺序输出0号节点所有节点的最短路径的长度

**样例输入：6**

**0 1 4 10000 10000 10000**

**1 0 2 7 5 10000**

**4 2 0 10000 1 10000**

**10000 7 10000 0 3 2**

**10000 5 1 3 0 6**

**10000 10000 10000 2 6 0**

**样例输出：0 1 3 7 4 9**

**拟解决关键问题：本题要解决的关键就是考虑利用什么算法来求出最短路径的长度。**

**二 概要设计**

首先使用二维数组来用邻接矩阵存储图。然后使用floyed算法来求出最短路径的长度。

**三 详细设计**

算法思想：

1，从任意一条单边路径开始。所有两点之间的距离是边的权，如果两点之间没有边相连，则权为无穷大。

2，对于每一对顶点 u 和 v，看看是否存在一个顶点 w 使得从 u 到 w 再到 v 比已知的路径更短。如果是更新它。

把图用邻接矩阵G表示出来，如果从Vi到Vj有路可达，则G[i][j]=d，d表示该路的长度；否则G[i][j]=无穷大。定义一个矩阵D用来记录所插入点的信息，D[i][j]表示从Vi到Vj需要经过的点，初始化D[i][j]=j。把各个顶点插入图中，比较插点后的距离与原来的距离，G[i][j] = min( G[i][j], G[i][k]+G[k][j] )，如果G[i][j]的值变小，则D[i][j]=k。在G中包含有两点之间最短道路的信息，而在D中则包含了最短通路径的信息。

**四 使用说明、测试分析及结果**

1. **说明如何使用你编写的程序；**

先输入一个小于等于100的正整数n，然后输入赋权图的邻接矩阵（10000表示无穷大，并且任意一条简单路径的长度都小于10000）

1. **测试结果与分析；**

**输入：6**

**0 1 4 10000 10000 10000**

**1 0 2 7 5 10000**

**4 2 0 10000 1 10000**

**10000 7 10000 0 3 2**

**10000 5 1 3 0 6**

**10000 10000 10000 2 6 0**

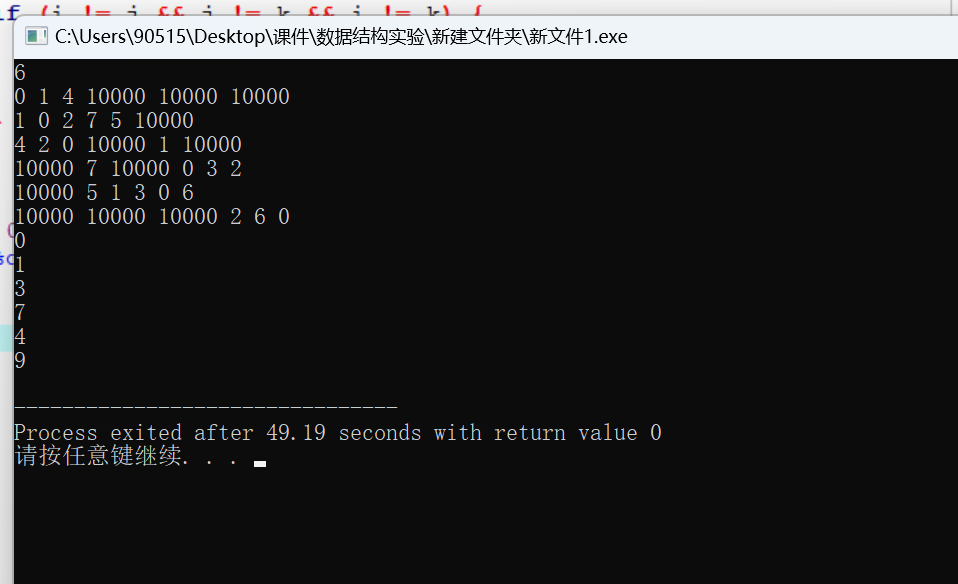
**输出：0 1 3 7 4 9**

**与样例相符，再次测试多组样例结果正确。**

1. **调试过程中遇到的问题是如何解决提以及对设计与实现的回顾讨论和分析**

思考是跑多遍dinkstra或者spfa还是用floyed算法，思考再三发现floyed的复杂度在此种情况下相对较优。于是选择floyed

**4、运行界面**



**五、实验总结**

实际动手操作，使用了Floyd算法处理多源最短路径问题。理解其复杂度。

中间涉及邻接矩阵存储无向图的知识点。

也加强了自身debug找错误的调试代码能力。

《数据结构》实验报告T009

班 级: DL062138 姓 名: 张苏宇 学 号: 2021302853

E-mail:905159071@qq.com 日 期: 2022/6/4

**实验4.2：**

**◎实验题目:**

**用迪杰斯特拉算法求赋权图中的最短路径**

**◎实验内容：**

**用迪杰斯特拉算法求一点到其余所有结点的最短路径。**

1. **需求分析**

**输入**：先输入一个小于100的正整数n，然后输入图的邻接矩阵（10000表示无穷大，即两点之间没有边），最后输入两个0到n-1的整数表示两个点。

**输出：**先用迪杰斯特拉算法求给定的第一个点到其余所有节点的最短路径。然后再输出给定的两个点之间的最短路径（按顺序输出最短路径上的每个点，每个数据占一行）

**样例输入：4**

**0 2 10 10000**

**2 0 7 3**

**10 7 0 6**

**10000 3 6 0**

**0 2**

**样例输出：0 1 2**

**拟解决关键问题：本题要解决的关键问题，如何跑dijkstra以及如何打印出路径。**

**二 概要设计**

首先我们需要审题，这里要求的是两个点之间的最短路径，而不是最短路径的长度，而裸的dijkstra算法用于解决单源最短路长度问题，也就是一个点到其他所有点的最短路径的长度，而求路径需要加一些优化，整体思路是用二维数组存储邻接矩阵，然后用一个pre来求出每个点的前驱，最终输出最短路径的长度。

**三 详细设计**

1. diklstra最短路，void dijkstra(int s)

首先我里面用到了dis，vis分别标记s点到到节点i的最短路径长度，以及节点i是否用过。

1. 初始时令 S={V0},T=V-S={其余顶点}，T中顶点对应的距离值 [1]

若存在<V0,Vi>，d(V0,Vi)为<V0,Vi>弧上的权值 [1]

若不存在<V0,Vi>，d(V0,Vi)为∞ [2]

2. 从T中选取一个与S中顶点有关联边且权值最小的顶点W，加入到S中 [1]

3. 对其余T中顶点的距离值进行修改：若加进W作中间顶点，从V0到Vi的距离值缩短，则修改此距离值 [1]

重复上述步骤2、3，直到S [1] 中包含所有顶点，即W=Vi为止

不过在上述过程中，我在更新到i点路径长度的时候，用一个pre数组将i的前驱记录，用于输出路径。

1. 输出路径 void PrintPath(int u,int v)

这个函数利用递归的思想，从尾部v一直向前递归，到达u后，向后递归输出最短路径。

**四 使用说明、测试分析及结果**

1. **说明如何使用你编写的程序；**

先输入一个小于100的正整数n，然后输入图的邻接矩阵（10000表示无穷大，即两点之间没有边），最后输入两个0到n-1的整数表示两个点。

1. **测试结果与分析；**

**输入：4**

**0 2 10 10000**

**2 0 7 3**

**10 7 0 6**

**10000 3 6 0**

**0 2**

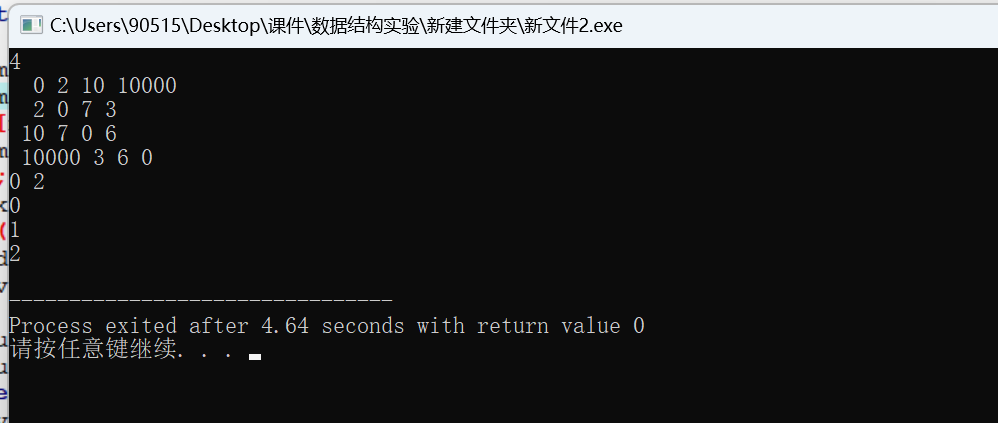
**输出：0 1 2**

**与样例相符，再次测试多组样例结果正确。**

1. **调试过程中遇到的问题是如何解决提以及对设计与实现的回顾讨论和分析**

在写题过程中因为考虑需要输出路径，但是我是直接记录的前驱，所以输出的路径是反着的，思考后我利用了递归的特性，成功的将路径反向输出，达到了我的目的。

1. **运行界面**



1. **实验总结**

通过此实验我学会了如何使用dijkstra算法求单源最短路长度。

并解决了输出路径求出结果的问题。（设置一个标记存储路径的数组）

《数据结构》实验报告T010

班 级: DL062138 姓 名: 张苏宇 学 号: 2021302853

E-mail:905159071@qq.com 日 期: 2022/6/4

**实验4.3：**

**◎实验题目:**

**用弗洛伊德算法求赋权图的两点间的最短路径的长度**

**◎实验内容：**

**用弗洛伊德算法求任意两点间的最短路径的长度**

1. **需求分析**

**输入**：先输入一个小于等于100的正整数n，然后输入图的邻接矩阵（10000表示无穷大，即两点之间没有边），之后再输入一个小于100的正整数m，最后的m行每行输入两个不同的0到n-1之间的整数表示两个点。

**输出：**用弗洛伊德算法求任意两点间的最短路径的长度，并输出这些两个点之间的最短路径的长度。

**样例输入：4**

**0 2 10 10000**

**2 0 7 3**

**10 7 0 6**

**10000 3 6 0**

**2**

**0 2**

**3 0**

**样例输出：9**

**5**

**拟解决关键问题：本题要解决的关键就是考虑利用什么算法来求出最短路径的长度。**

**二 概要设计**

首先使用二维数组来用邻接矩阵存储图。然后使用floyed算法来求出最短路径的长度。

**三 详细设计**

**1）算法思想：**

可以将问题分解:

从任意节点i到任意节点j的最短路径不外乎2种可能，1是直接从i到j，2是从i经过若干个节点k到j。所以，我们假设Dis(i,j)为节点u到节点v的最短路径的距离，对于每一个节点k，我们检查D(i,k) + Dis(k,j) < Dis(i,j)是否成立，如果成立，证明从i到k再到j的路径比i直接到j的路径短，我们便设置Dis(i,j) = Dis(i,k) + Dis(k,j)，这样一来，当我们遍历完所有节点k，Dis(i,j)中记录的便是i到j的最短路径的距离。

**2).算法描述：**

a.从任意一条单边路径开始。所有两点之间的距离是边的权，如果两点之间没有边相连，则权为无穷大。

b.对于每一对顶点 u 和 v，看看是否存在一个顶点 w 使得从 u 到 w 再到 v 比己知的路径更短。如果是更新它。

给出矩阵，其中矩阵A是邻接矩阵，而矩阵Path记录u,v两点之间最短路径所必须经过的点。

先用邻接矩阵G把图表示出来:

如果从Vi到Vj有路可达，则G[i][j]=d，d表示该路的长度；

否则G[i][j]=无穷大。

用矩阵D用来记录所插入点的信息，D[i][j]表示从Vi到Vj需要经过的点，初始化D[i][j]=j。

将图中的各个点放进去，然后比较现在和原来的距离，G[i][j] = min( G[i][j], G[i][k]+G[k][j] )，如果G[i][j]的值变小，则D[i][j]=k。在G中包含有两点之间最短道路的信息，而在D中则包含了最短通路径的信息。

**3）Floyd算法过程矩阵的计算----十字交叉法**

方法：两条线，从左上角开始计算一直到右下角

**四 使用说明、测试分析及结果**

1. **说明如何使用你编写的程序；**

先输入一个小于等于100的正整数n，然后输入赋权图的邻接矩阵（10000表示无穷大，并且任意一条简单路径的长度都小于10000）

1. **测试结果与分析；**

**输入：4**

**0 2 10 10000**

**2 0 7 3**

**10 7 0 6**

**10000 3 6 0**

**2**

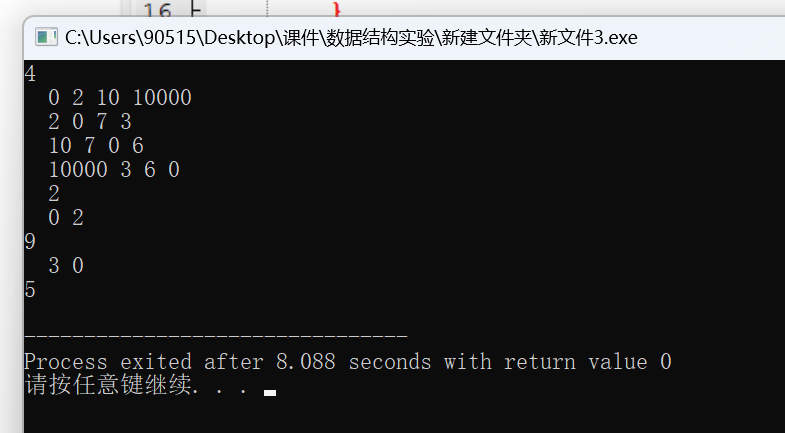
**0 2**

**3 0**

**输出：9 5**

**与样例相符，再次测试多组样例结果正确。**

1. **运行界面**



**五、实验总结**

通过此实验学习了如何用floyed求最短路径的长度。

《数据结构》实验报告T011

班 级: DL062138 姓 名: 张苏宇 学 号: 2021302853

E-mail:905159071@qq.com 日 期: 2022/6/4

**实验4.4：**

**◎实验题目:**

**用弗洛伊德算法求赋权图中任意两点间的最短路径**

**◎实验内容：**

**用弗洛伊德算法求任意两点间的最短路径，并输出指定的m对节点间的最短路径。**

1. **需求分析**

**输入**：先输入一个小于等于100的正整数n，然后输入赋权图的邻接矩阵（10000表示无穷大，之后再输入一个小于100的正整数m，最后的m行每行输入两个不同的0到n-1之间的整数表示两个点。

**输出：**用弗洛伊德算法求任意两点间的最短路径，并输出这些两个点之间的最短路径

**样例输入：**4

0 2 10 10000

2 0 7 3

10 7 0 6

10000 3 6 0

2

0 2

3 0

**样例输出：**0 1 2 3 1 0

**拟解决关键问题：**

**二 概要设计**

使用二维数组来用邻接矩阵存储图。然后使用floyed算法来求出最短路径的长度。不过这里因为不仅需要最短路径的长度，还需要最短路径，所以需要加入一个二维数组来当存储中转点，以此来输出最短路径。

**三 详细设计**

**1.求最短路以及为输出路径准备**

floyed算法是每次判断一下用k来做中转点的时候，是否能够使得i，j之间的路径最短，我们需要输出最短路径就需要记录k，我们利用一个二维path数组，path[i][j]用来记录i，j之间的中转点。

核心代码如下：

for(int k=0;k<n;k++)

{

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

if(i!=j && j!=k && i!=k)

{

if(map[i][j] > map[i][k] + map[k][j])

{

map[i][j] = map[i][k] + map[k][j];

path[i][j] = k;

}

}

}

}

}

**2.利用函数输出最短路**

我们考虑可以利用递归输出，我们需要求的u,v之间的最短路径，我们可以输出u,path[u][v] 之间最短路，然后输出 path[u][v],v之间的最短路，一直这样递归到某条边与u或者v直接相连。

void PrintPath(int u,int v)

{

int k = path[u][v];

if(path[u][v] == -1) return ;

PrintPath(u,k);

printf("%d\n",k);

PrintPath(k,v);

}

**四 使用说明、测试分析及结果**

**1.说明如何使用你编写的程序；**

先输入一个小于等于100的正整数n，然后输入赋权图的邻接矩阵（10000表示无穷大，之后再输入一个小于100的正整数m，最后的m行每行输入两个不同的0到n-1之间的整数表示两个点。

**2.测试结果与分析；**

**输入：**4

0 2 10 10000

2 0 7 3

10 7 0 6

10000 3 0 6 0

2

0 2

3 0

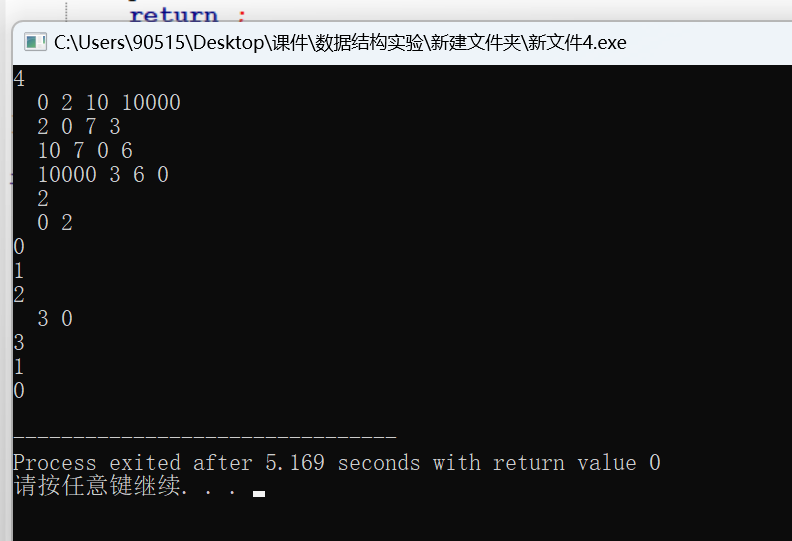
**输出：**0 1 2 3 1 0

**与样例相符，再次测试多组样例结果正确。**

1. **调试过程中遇到的问题是如何解决提以及对设计与实现的回顾讨论和分析**

一开始我只是输出了path[u][v] 和v之间的最短路径，却没有考虑u和path[u][v]之间的中转点，这导致少输出了一部分中转点，思考后得以解决。

**4.运行界面**



**五、实验总结**

通过此实验再次复习重温加强记忆学会了如何用floyed算法处理多源最短路问题以及如何利用floyed算法打印出最短路的路径。

并加强了自身debug调试代码的能力，以及对算法复杂度的评估能力，是对理论知识记忆的一次强化理解。