**北京科技大学实验报告**

学院：计算机与通信工程学院 专业：通信工程 班级：通信1701

姓名：胡成成 学号：41724260 实验日期： 2020年 4月 6日

**实验名称：F算法函数实现**

**实验目的：**

1. 进一步熟悉F算法中的矩阵操作流程；能查询任意两点间的最短距离和路由。
2. 熟悉MATLAB以块操作为中心的编程思想。

**实验仪器：**

1. 操作系统：WIN10
2. 操作软件：MATLAB R2018b

**实验原理：**

Floyd算法是一个经典的动态规划算法，它又被称为插点法。该算法名称以创始人之一、1978年图灵奖获得者、斯坦福大学计算机科学系教授罗伯特·弗洛伊德命名。Floyd算法是一种利用动态规划的思想寻找给定的加权图中多源点之间最短路径的算法，算法目标是寻找任意两点的最短路径。

算法的基本原理过程：

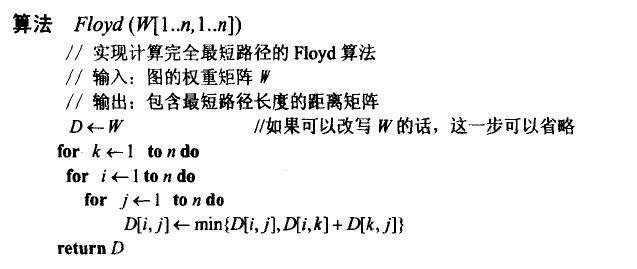
1. 从任意一条单边路径开始。所有两点之间的距离是边的权，如果两点之间没有边相连，则权为无穷大。
2. 对于每一对顶点 u 和 v，看看是否存在一个顶点 w 使得从 u 到 w 再到 v 比已知的路径更短。如果是更新它。

**实验内容与步骤：**

1. 梳理算法的基本原理与过程
2. 编写程序的伪代码逻辑
3. 创建MATLAB函数文件将算法逻辑实现
4. 创建路径可视化函数清楚路由过程
5. 编写主程序设置初始距离矩阵测试算法

**程序说明及流程图：**

1. 算法实现流程：



1. Flody算法代码实现过程：
   1. 定义函数的输入输出，其中a为初始输入距离矩阵，d为最终输出的距离矩阵，r为最终输出的路由矩阵：

function [d,r]=floyd(a)

* 1. 初始化距离矩阵和路由矩阵：

n=size(a,1);

d=a; % 初始化距离矩阵

% 初始化路由矩阵

for i=1:n

for j=1:n

r(i,j)=j;

end

end

* 1. Flody算法，遍历所有节点，更新距离矩阵和路由矩阵：

for k=1:n

for i=1:n

for j=1:n

if d(i,k)+d(k,j)<d(i,j)

d(i,j)=d(i,k)+d(k,j);

r(i,j)=r(i,k);

end

end

end

end

* 1. 编写路径展示函数

function DisplayPath(route, start, dest)

% 打印出任意两点之间的最短路径

% route : 路由表

% start : 起点index

% dest : 终点index

i = 1;

while 1

if(route(start, dest) ~= dest)

fprintf('V%s -> ', num2str(start));

start = route(start, dest);

else

fprintf('V%s -> ', num2str(start));

fprintf('V%s\n', num2str(dest));

break;

end

end

* 1. 设计初始距离矩阵数据

a = [0 2 6 4;

inf 0 3 inf;

7 inf 0 1 ;

5 inf 12 0];

* 1. 运行算法和展示路径

[d,r]=floyd(a)

disp('--------------------------')

for i = 1 : 4

for j = 1 : 4

DisplayPath(r, i, j);

end

end

**实验结果与分析：**

1. 最终的距离矩阵和路由矩阵输出：

d =

0 2 5 4

9 0 3 4

6 8 0 1

5 7 10 0

r =

1 2 2 4

3 2 3 3

4 4 3 4

1 1 1 4

1. 遍历所有路由线路：

V1 -> V1

V1 -> V2

V1 -> V2 -> V3

V1 -> V4

V2 -> V3 -> V4 -> V1

V2 -> V2

V2 -> V3

V2 -> V3 -> V4

V3 -> V4 -> V1

V3 -> V4 -> V1 -> V2

V3 -> V3

V3 -> V4

V4 -> V1

V4 -> V1 -> V2

V4 -> V1 -> V2 -> V3

V4 -> V4

**遇到的问题：**

由于之前参加过数学建模比赛，该算法已经学习并实现应用过，此次实验并没有遇到什么大问题。

**实验总结：**

通过本次实验，将课本理论的Flody算法通过编程实践，理论与实践结合，更有利于课上对算法逻辑的熟练掌握。此次实验也再次加固了图论的知识，有利于之后对新算法的学习与实践。

**附录（程序源代码）:**

1. **Floyd.m函数**

% floyd.m

% 采用floyd算法计算图a中每对顶点最短路

% d是矩离矩阵

% r是路由矩阵

function [d,r]=floyd(a)

n=size(a,1);

% 初始化距离矩阵

d=a;

% 初始化路由矩阵

for i=1:n

for j=1:n

r(i,j)=j;

end

end

r;

% Floyd算法开始

for k=1:n

for i=1:n

for j=1:n

if d(i,k)+d(k,j)<d(i,j)

d(i,j)=d(i,k)+d(k,j);

r(i,j)=r(i,k);

end

end

end

k;

d;

r;

end

d

r

1. **DisplayPath.m函数**

% DisplayPath.m 打印路径函数

function DisplayPath(route, start, dest)

% 打印出任意两点之间的最短路径

% route : 路由表

% start : 起点index

% dest : 终点index

i = 1;

while 1

if(route(start, dest) ~= dest)

fprintf('V%s -> ', num2str(start));

start = route(start, dest);

else

fprintf('V%s -> ', num2str(start));

fprintf('V%s\n', num2str(dest));

break;

end

end

1. **Main.m主程序**

a = [0 2 6 4;

inf 0 3 inf;

7 inf 0 1 ;

5 inf 12 0];

[d,r]=floyd(a)

disp('--------------------------')

for i = 1 : 4

for j = 1 : 4

DisplayPath(r, i, j);

end

end