|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《数据结构》实验报告** |
|  |
| 实验四  图形结构及其应用  学 院: 计算机科学与技术   |  |  | | --- | --- | | 姓 名: | 王铭 | | 学 号: | 190110509 | | 专 业: | 计算机科学与技术 | | 日 期: | 2020-05-09 | |

# 一、问题分析

实验一：用图的一个顶点代表一个人，图的边的权代表两人关系的亲近程度，计算出从某一点出发到各个顶点的最短距离。

实验二：根据输入的输入层神经元权值根据给定的公式计算各个神经元的权值，最后输出输出层神经元的权值。

# 二、详细设计

## 2.1 设计思想

实验一：采用Dijkstra算法，用dis向量存储从顶点k出发到各点的最短路径长度，初值赋为k到各点边的权，做n-1次循环，每次找到dis中的最小值并标记已找到最短路径，找到dis中未找到最短路径的点比较从找到的点到各点的距离与dis中存储的最小路径值，若前者更小，则修改dis中存储的最小路径值，最后将结果保留在ans向量中。

实验二：通过输入层神经元的权值，计算与输出层神经元相连的神经元的权值，并将计算出来的神经元视为输出层神经元继续计算剩下的神经元的权值，最后输出输出层神经元的权值。

## 2.2 存储结构及操作

(1) 存储结构

struct Edge

{

int start, end, weight;

int next;

};

struct Graph

{

struct Edge edegs[maxm];

int head[maxn];

int tot;

};

(2) 涉及的操作

实验一：

a) init\_graph函数：初始化有向图并返回指向该图的指针。

b) add\_edge函数：向有向图中添加边。

c) visit\_graph函数：遍历以各个顶点为起点的边。

d)get\_path函数：计算从顶点k出发到各个顶点的最短路径长度。

实验二：

a) init\_graph函数：初始化有向图并返回指向该图的指针。

b) add\_edge函数：向有向图中添加边。

c) visit\_graph函数：遍历以各个顶点为起点的边。

d) Panduan函数：判断各点的权值是否已计算结束。

e) forward\_propagation函数：计算输出层神经元的权值。

## 2.3 程序整体流程

整体流程：

实验一:如图1所示

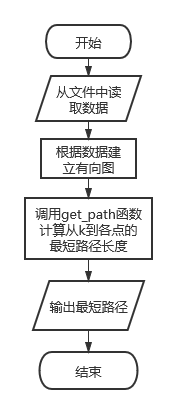


图1

实验二：如图2所示



图2

forward\_propagation函数：如图3所示

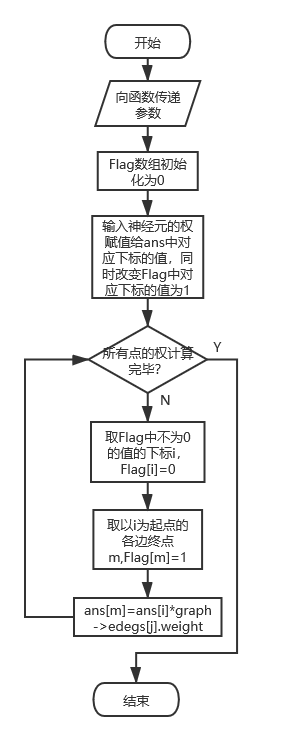


图3

get\_path函数：如图4所示

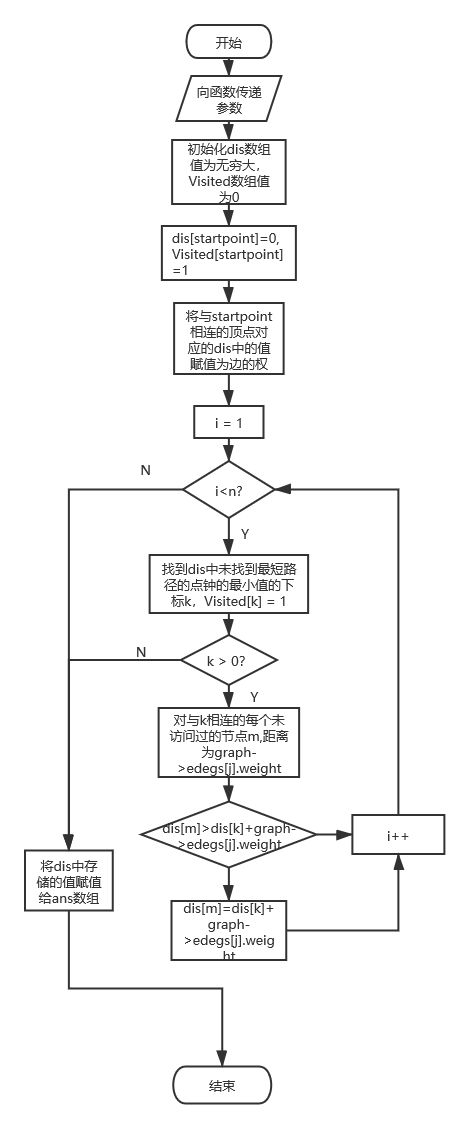


图4

# 三、用户手册

1）通过文件进行输入，文件内的数据格式应为：

i)实验一：

第一行包含三个整数，分别表示点的个数、有向边的个数、出发点的编号。

接下来行每行包含三个整数，分别表示第条有向边的出发点、目标点和亲疏值。

ii)实验二：

第一行包含四个正整数，表示总神经元个数，总的边个数，输入层神经元个数以及输出层神经元个数。

接下来行，每三个正整数，表示有一条从号神经元指向号神经元的有向边，边权为。

接下来一行个正整数，表示每个输入层神经元的输入值。

注：到号节点为输入层神经元，到号节点为隐藏层神经元，到号节点为输出层神经元。

2）程序输出的格式为：

i)实验一：

一行，包含个用空格分隔的整数，其中第个整数表示从点出发到点的亲疏值（若则为0，若从点无法到达点，则亲疏值为2147483630）。

ii)实验二：

输出一行包含个整数，表示每个输出层神经元的输出值。

# 四、结果

实验一如图5所示：

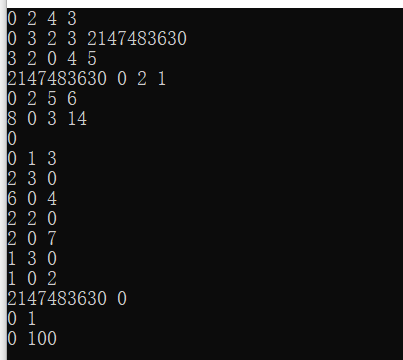


图5

实验二如图6所示：

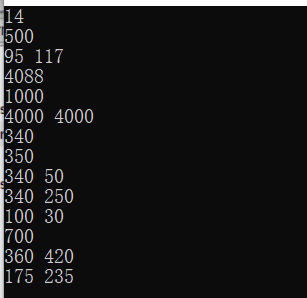


图6

# 五、总结

该实验涉及到的数据结构和算法，以及遇到的问题和收获。

该实验中，使用向前星存储结构存储图，应用了Dijkstra算法，实现了求解最短路径问题。用类似拓扑排序的算法解决了求解输出神经元的权值。通过本次实验，我能够更加熟悉地使用不同的存储结构实现图的存储，掌握了图的基本操作，同时更加深入地理解了Dijkstra算法及拓扑排序算法，但在实验课时考虑一些特殊情况时仍有欠缺，导致输入某些特殊数据时无法得到正确结果，仍需加强对细节的处理。