

数据结构与算法基础课程实验报告

实验 3：图及其应用

姓名	郭梓贤	院系	计算机科学与技术学院	学号	1170500112	
任课教师	臧天仪		指导教师	臧天仪		
实验地点	格物楼 207		实验时间	周六 13:30-15:30		
实验课表现	出勤、表现得分 10%		实验报告 得分 40%		实验总分	
	操作结果得分 50%					

实验目的：

1. 掌握有向图的建立

2. 掌握有向图中各种重要性质在解决实际问题中的应用。

3. 掌握 Floyd 算法

4. 掌握 Dijkstra 算法

实验内容：

1 本次实验从文件 ex3_Graph1.txt，ex3_Graph2.txt 读取数据文件建立有向网，并遍历图，输出节点值；注意数据文件 矩阵中 -1 表示 不存在这条有向边，例如 ex3_Graph1.txt 中，s0 行 s3 列 的元素-1，表示不存在从 s0 到 s3 的边。

2.针对上步建立的第一个图的有向网,分别求出从节点 s0 到节点 s1,s2,s3,s4 的最短路径及对应的权值；针对上步建立的第二个图，求出节点 a 到节点 h 的最短路径及对应权值。

实验要求：（学生对预习要求的回答）（5 分）

得分：

图的存储结构有三种：邻接矩阵、邻接表、十字链表

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。通过一个图的权值矩阵求出它的每两点间的最短路径矩阵。从图的带权邻接矩阵 $A=[a(i,j)]\ n\times n$ 开始，递归地进行n次更新，即由矩阵 $D(0)=A$ ，按一个公式，构造出矩阵 $D(1)$ ；又用同样地公式由 $D(1)$ 构造出 $D(2)$ ；……最后又用同样的公式由 $D(n-1)$ 构造出矩阵 $D(n)$ 。矩阵 $D(n)$ 的i行j列元素便是i号顶点到j号顶点的最短路径长度，称 $D(n)$ 为图的距离矩阵，同时还可引入一个后继节点矩阵path来记录两点间的最短路径。采用松弛操作，对在i和j之间的所有其他点进行一次松弛。时间复杂度为 $O(n^3)$;

Dijkstra算法主要特点是以起始点为中心向外层层扩展，直到扩展到终点为止。通过每一轮的“松弛”维护一个表示每个顶点到出发点的最短路径长度的数组，最终获得最短路径。时间复杂度为 $O(n^2)$.

实验过程中遇到的问题如何解决的？（10 分）（着重从软件调试、质量保证、结果分析方面进行阐述）	得分：
<p>问题 1：</p> <p>实验提供了输入数据文件，需要从中提取建立有序图的信息。采用 fgets 的输入方式，获取字符串，然后通过访问特定位来获得相应数据。</p> <p>建立有向图采用邻接矩阵的表示方法，因为数据提供方式也是以矩阵的形式。第一行字符串读取可以获得总的顶点数，因为读取的是数字的字符表示，需要-48 变化成真正的顶点数，然后依据顶点数建立矩阵，进而读取输入，构建邻接矩阵。</p> <p>对于第二组数据同理，但是需要注意的是，邻接矩阵记录了定点之间的关系，第一个图中，顶点命名直接通过数字，而第二个图的顶点另有命名，所以多出一个一维数组存储各个顶点的信息。</p> <p>问题 2：</p> <p>因为需求是从某个点出发的最短路径，所以采用 Dijkstra 算法。</p> <p>目的是维护一个记录每个顶点到 s0 的最短路径的一维数组，和一个记录最短路径中每个顶点前一个顶点的一维数组。Dijkstra 算法的操作方法是，每一轮从当前最短路径数组中，找到最短路径长度和最短路径的顶点，对所有未确定最短路径的顶点的距离进行“松弛”，最终完成对所有定点的最短路径的记录。顶点是否已经确定了到 s0 的最短路径采用一个一维标记位数组进行记录。</p> <p>需要注意的是，Dijkstra 算法不能解决负权边的问题，所以之前建立的邻接矩阵的-1 项需要处理，将-1 改成一个较大的正数，使得算法能正常找到最短路。</p> <p>最后输出通过循环访问记录最短路径前一个顶点的数组和记录最短路径长度的数组就可以获得最短路径和对应的路径长度。</p>	
本次实验的体会（结论）（10 分）	得分：
<p>Dijkstra 算法采用了贪心的思想，每次查找与出发点距离最近的点，因此，它不能用来解决负权边的问题，所以此例中邻接矩阵中的-1 项需要进行特殊处理，此处处理方式为其改成一个大的正数。</p>	
思考题：（15 分）	
思考题 1：（5 分）	得分：
<p>思考题 1：Floyd 的时间复杂度是什么？为什么？</p> <p>复杂度为 $O(n^3)$，Floyd 算法包含三重循环，每次松弛需要计算各个顶点到其他顶点的最短距离，是两重循环，复杂度为 n^2。再考虑到每一轮的松弛都要多考虑一个顶点，所以一共需要进行 n 轮松弛，所以最终的复杂度为 n^3。</p>	
思考题 2：（5 分）	得分：

思考题 2：在图的表示中通常都有哪几种方法？其特点是什么？

邻接矩阵和邻接表两种。

邻接矩阵通过二维数组中某元素的值来确定此元素两个下标索引值所对应的顶点的关系，并且将所有点与点之间的关系（即使没有边）都表示出来了。如果是无向图，则邻接矩阵关于对角线对称，有向图的邻接矩阵不一定关于对角线对称。

邻接矩阵适合对稠密图进行表示。

邻接表只把存在关系的点表示出来。建立由每个顶点组成的单链表，然后将邻接自每个顶点的顶点使用链表进行连接。适合对稀疏图进行表示。

思考题 2：（5 分）

得分：

思考题 3：图的存储结构有哪些？并简要说明其优缺点？

图的存储结构有三种：邻接矩阵、邻接表、十字链表

邻接矩阵适合对存储稠密图的信息。在获取任意两个顶点之间的边或弧的信息时比较方便，但是可能造成存储空间浪费。

邻接表适合存储稀疏图的信息，图比较稀疏时，邻接表占用空间比邻接矩阵少，但要获取有向图中任意两顶点之间的关系，需要遍历两个链表，较为麻烦。而且如果要求解某一点的入度，则需要遍历所有的链表。

十字链表可以看作将有向图的邻接表和逆邻接表相结合的储存结构，能够弥补邻接表中不易寻找顶点入度的缺点，但操作比较繁琐。

指导教师特殊评语：

指导教师签字：

日期：