

哈爾濱工業大學

数据结构实验报告

实验三

专 业 计算机类

学 号 120L022314

学 生 瞿久尧

指 导 教 师 苗东菁老师

一. 简介/问题求解思路

1. 问题：哈尔滨的某个乡镇里有 n 个村庄，分别用 1 到 n 表示。为了让各个村庄之间实现互联互通，当地政府决定为这个 n 个村庄修公路，让任何一个村庄都可以经过公路到达另一村庄。直接在所有村庄之间都修一条路花费太大，所以当地政府想要请你来帮他们想一个办法：如何用最少的钱来修路？请你告诉他们修路最少需要用多少钱，并且给出修建方案？
2. 求解思路：在无向图中依据权值选取最小生成树，权值加和即修路的钱，寻找最小生成树的过程，即在村庄间修路。

二. 数据结构设计与实现

逻辑结构：堆（斐波那契堆），邻接表

存储结构：链表，数组

实现

首先输入村庄个数 n 即修路方案数 m ，根据这 m 个修路方案可建立起图的邻接表，根据邻接表建立起一个堆（斐波那契堆），利用堆操作实现 Prime 算法

三. 算法设计及分析

1. 函数设计

1.1 Initial() 函数，初始化图，使图上的所有点到最小生成树的距离为 MAX（先定义 MAX 为无穷大），并且为图上的顶点编序。

1.2 Insert() 函数，根据用户输入将有路的村庄加入邻接表中，并修改权值为建路需要花的钱。

1.3 Dui() 函数，（参考算法导论）建立一个堆（斐波那契堆），并且包含一些堆的操作：

1.3.1 MAKE-HEAP()，创建并返回一个新的不含任何元素的堆；

1.3.2 EXTRACT-MIN(H)，从堆 H 中删除最小关键字的元素，并返回一个指向该元素的指针；（用于找到最便宜的修路方案后，将该方案去掉，便于后续操作）

1.3.3 UNION(H1, H2)，创建并返回一个包含堆 H1 和堆 H2 中所有元素的新堆。堆 H1 和堆 H2 由这一操作“销毁”；（用于维护堆，减少时间复杂度）

1.3.4 DECREASE-KEY(H, x, k)，将堆中元素 x 的关键字赋予新值 k。（用于在寻找最小生成树的过程中不断更改堆中各元素的值，即各顶点到

最小生成树的距离)

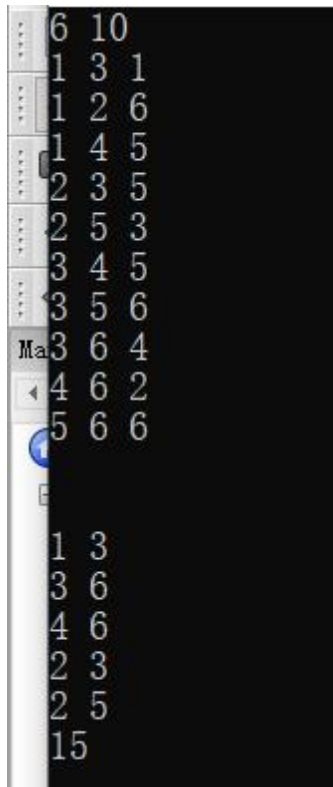
2. 算法分析

2.1 空间复杂度

其中有数个记录顶点的数组和链表，还有数个记录路的数组和链表，因此空间复杂度应为 $O(n+m)$

2.2 时间复杂度，普通的 Prime 算法的时间复杂度为 $O(n^2)$ ，利用堆优化后时间复杂度能节省到 $O(m \lg n)$ ，而利用斐波那契堆优化后时间复杂度可达到 $O(m+n \lg n)$

四. 实验测试结果及结果分析



结果分析：1. 上部分为输入，下部分为输出。

2. 输入时首先需要输入村庄数量 n 及可供修路的方案数 m 。然后每行输入的三个数，前两个数为两个村庄，第三个数为在这两个村庄间修路的价格，共 m 行。

3. 输出时首先给出修路方案，在哪两个村庄间修路，最后给出总价格，即这些路的权值加和。

五. 实验中遇到的问题及解决方法

问题：建堆的操作以及堆中需要的一些基本操作在实现程度上并不完善

解决：参考《算法导论》，并利用讲解的斐波那契堆对算法进行优化

参考文献

《算法导论》