

**数据结构实验报告**

**实验三**

专 业 计算机科学与技术

学　　 号 1190201215

学 生 冯开来

指 导 教 师 苗东菁

1. **简介/问题求解思路**

**Problem 1.1**

Prim算法是求最小生成树的算法，其基本思想是：从一定点出发，找出此点邻接边中最短的边加入对应顶点，再次寻找这两条边的最短邻接边，依次类推，加入所有没有被访问过的边。

**Problem 1.2**

把n个顶点看成看成n棵分离的树（每棵树只有一个顶点），每次选取可连接两个分离树中权值最小的边把两个分离的树合成一个新的树取代原来的两个分离树，如果重复n-1步后便得到最小生成树。

**Problem 2**

设置并逐步扩充一个集合S，存放已求出其最短路径的顶点，则尚未确定确定最短路径的顶点集合是V-S，其中V为网中所有顶点集合。按最短路径长度递增的顺序逐个以V-S中的顶点加到S中，指导S中包含全部顶点，而V-S为空。

1. **数据结构设计与实现**

**Problem 1.1**

从某一个顶点开始（不妨假设从顶点0开始），标记它，并将它邻接表中的边全部加入到队列中；

从队列中选出并删除权值最小的那条边，先检查这条边的两个顶点是否都已经被标记过，若是，加入这条边会导致成环。跳过这条边，选择并删除下一个权值最小的边，直到某条边的两个顶点不是都被标记过，然后将其加入到MST中，将该边的另一个顶点标记，并将所有与这个顶点相邻且未被标记的顶点的边加入队列。

 重复上述步骤，直到列表中的元素都被删除。

上述代码，MST是一个队列，用来存放MST中的边（按照加入的顺序）。edges是一个优先队列，每次访问一个点，就将它的邻接边中，另一顶点未被标记的那些边加入。每次要从edges里选出权值最小的边删除。

**Problem 1.2**

遍历所有的边，将权值最小的边收入MST中，这里可以使用最小堆的数据结构，运用堆排序得到最小堆。

每一次将最小边收入MST时，需要检验在MST中新收录的边是否会和已有的边构成回路。这里运用并查集的数据结构，检查新加入的边两个顶点V1，V2是否在现有的最小生成树子集中构成回路。得到V1所属的连通集名称和V2连通集名称，若V1和V2连通，则该边不能要。

重复上述步骤，记录每条边的权重，直到边集为空。最后检查所有的边是否遍历过，若没有，则该图不是连通图。最后输出总权重。

**Problem 2**

类似Prim算法，不过此时的dist不是顶点到最小生成树MST的距离，而是顶点到源点的距离。

1. **算法设计及分析**

**Problem 1.1**

时间复杂度为

空间复杂度

**Problem 1.2**

时间复杂度为

空间复杂度为

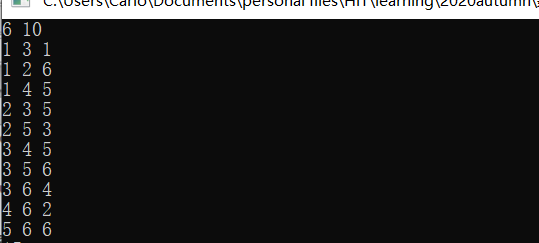
**Problem 1.3**

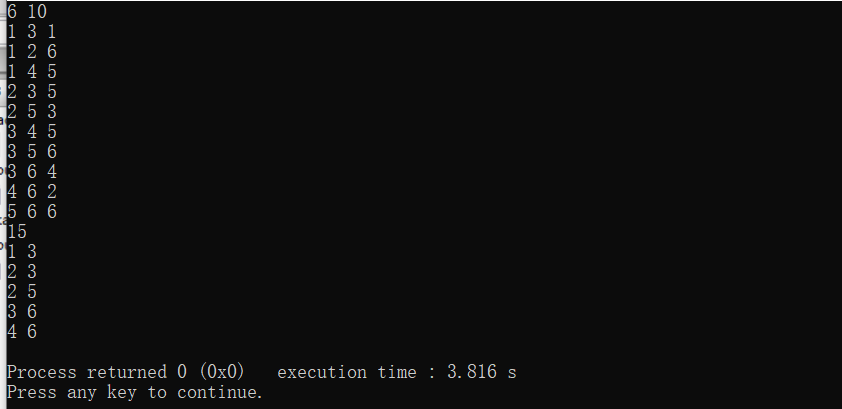
时间复杂度为

空间复杂度为

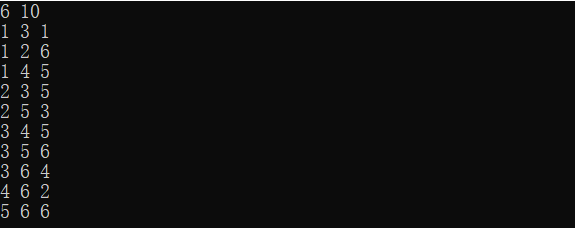
1. **实验测试结果及结果分析**

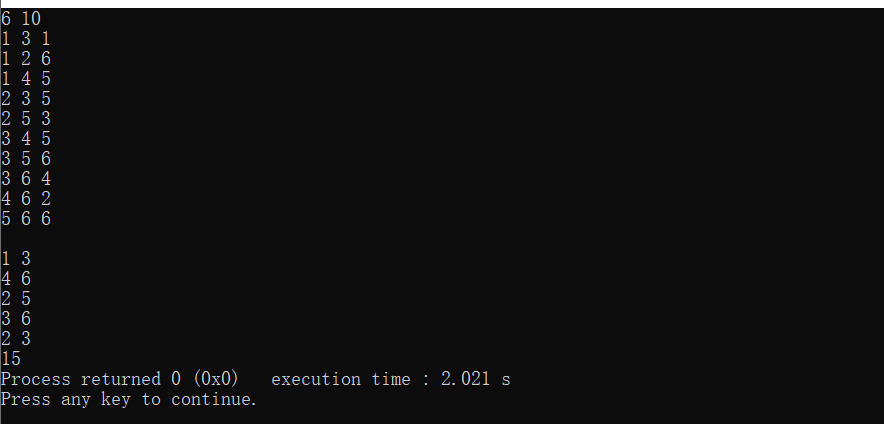
**Problem 1.1**



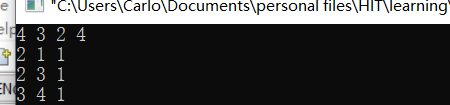
****

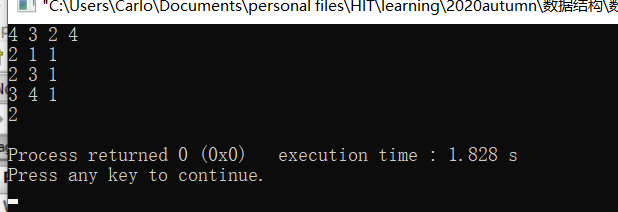
**Problem 1.2**

****

****

**Problem 3**

****

****

1. **实验中遇到的问题及解决方法**

**Problem 1.1**

在每一次寻找dist最小值的时候，这里运用的是遍历的方法，此时的时间复杂度为，如何减少时间复杂度？

可以使用最小堆的数据结构，将dist的值存入最小堆中，不过每次取最小dist的时候注意既不能是无穷大也不能是0。优化后的时间复杂度为。

**Problem 1.2**

在每一次寻找权重最小的边的时候，时间复杂度为，如何减小时间复杂度？

可以使用最小堆的方法，将边的权重存入最小堆中，这样时间复杂度可以缩减为。

在判断收录的新边是否会构成回路的时候，这里遇到困难，因为全部遍历肯定显得不切实际，所以这里使用了并查集的数据结构，时间复杂度为，如何优化？

并查集可以运用路径压缩和按值合并，使用路径压缩的话，每次可以直接找到祖先，不用一步步父亲节点的找，这样的时间复杂度为

**Problem 3**

类似Problem 1.1，在寻找dist最小值的时候可以使用最小堆来进行优化，这样时间复杂度就会减少到。

**参考文献**

无。