算法设计与分析实验报告

实验名称：最小生成树(Prim算法和Kruskal算法)

一、问题陈述，相关背景、应用及研究现状的综述分析

1.问题陈述：

在一个无向带权连通图G中寻找一个子图G′，使得G′是一棵包含G中所有顶点的同时所含权最小的树；

分别运用Prim算法和Kruskal算法求解最小生成树,并对其时间复杂度和空间复杂度进行分析；

2. 相关背景

普里姆算法（Prim算法）。图论中的一种算法，可在加权连通图里搜索最小生成树。意即由此算法搜索到的边子集所构成的树中。不但包括了连通图里的全部顶点（英语：Vertex (graph theory)），且其全部边的权值之和亦为最小。该算法于1930年由捷克数学家沃伊捷赫•亚尔尼克发现。并在1957年由美国计算机科学家罗伯特•普里姆（英语：Robert C. Prim）独立发现。1959年，艾兹格•迪科斯彻再次发现了该算法。因此，在某些场合，普里姆算法又被称为DJP算法、亚尔尼克算法或普里姆－亚尔尼克算法。

Kruskal算法是一种用来寻找最小生成树的算法，由Joseph Kruskal在1956年发表。用来解决同样问题的还有Prim算法和Boruvka算法等。三种算法都是贪婪算法的应用。和Boruvka算法不同的地方是，Kruskal算法在图中存在同样权值的边时也有效。

二、模型拟制、算法设计和正确性证明

1.算法设计：

Prim:

1).输入：一个加权连通图，其中顶点集合为V，边集合为E;

2).初始化：$V_{new} = \{x\}$，其中x为集合V中的任一节点(起始点)，$E_{new} = \{\}$,为空;

3).重复下列操作，直到 $V_{new} = V$：

a.在集合E中选取权值最小的边<u, v>，其中u为集合 V_{new} 中的元素，而v不在 V_{new} 集合当中，并且v∈V(如果存在有多条满足前述条件即具有相同权值的边，则可任意选取其中之一);

b.将v加入集合 V_{new} 中，将<u, v>边加入集合 E_{new} 中;

4).输出：使用集合 V_{new} 和 E_{new} 来描述所得到的最小生成树。

Kruskal:

1).记Graph中有v个顶点，e个边

2).新建图 $Graph_{new}$，$Graph_{new}$ 中拥有原图中相同的e个顶点，但没有边

3).将原图Graph中所有e个边按权值从小到大排序

4).循环：从权值最小的边开始遍历每条边 直至图Graph中所有的节点都在同一个连通分量中

if 这条边连接的两个节点于图 $Graph_{new}$ 中不在同一个连通分量中添加这条边到图 $Graph_{new}$ 中

三、时间和空间复杂性分析

Prim:

时间复杂度：

$$O(n^2)$$

空间复杂度：

需要申请并查集的空间，因此空间复杂度为 O（n）

Kruskal:

时间复杂度：

当图的边数为 e 时，Kruskal 算法所需的时间是 O（eloge）

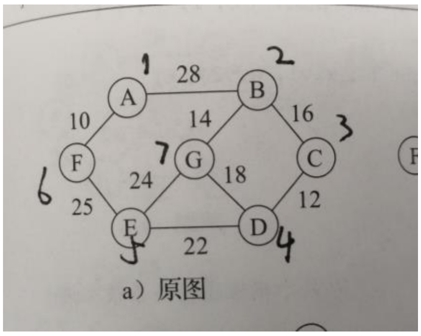
空间复杂度：

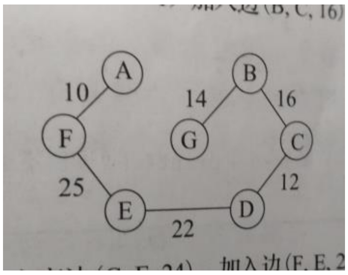
需要申请额外的数组来存储信息，空间复杂度为 O(n)

四、程序实现和实验测试过程

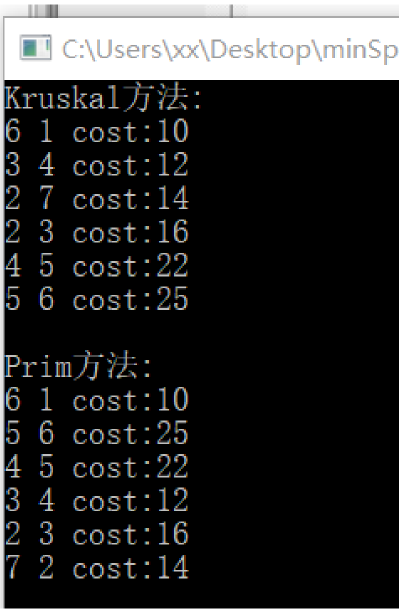
源程序见 MinSpanTree.cpp

测试过程：
结点以及权值数据





测试过程



五、总结