**实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | 《算法分析与设计》 | **实验日期** | 2021 年3月 8日 至2021 年 3 月 15日 | | | | |
| **学生姓名** | 王海轩 | **所在班级** | 计算机195 | | | **学号** | 2019212212194 |
| **实验名称** | 采用Kruskal算法构造最小生成树 | | | | | | |
| **实验地点** | 勤园13-208 | | | **同组人员** |  | | |

# 问题

[描述算法问题，首选形式化方式（数学语言），其次才是非形式化方式（日常语言）]

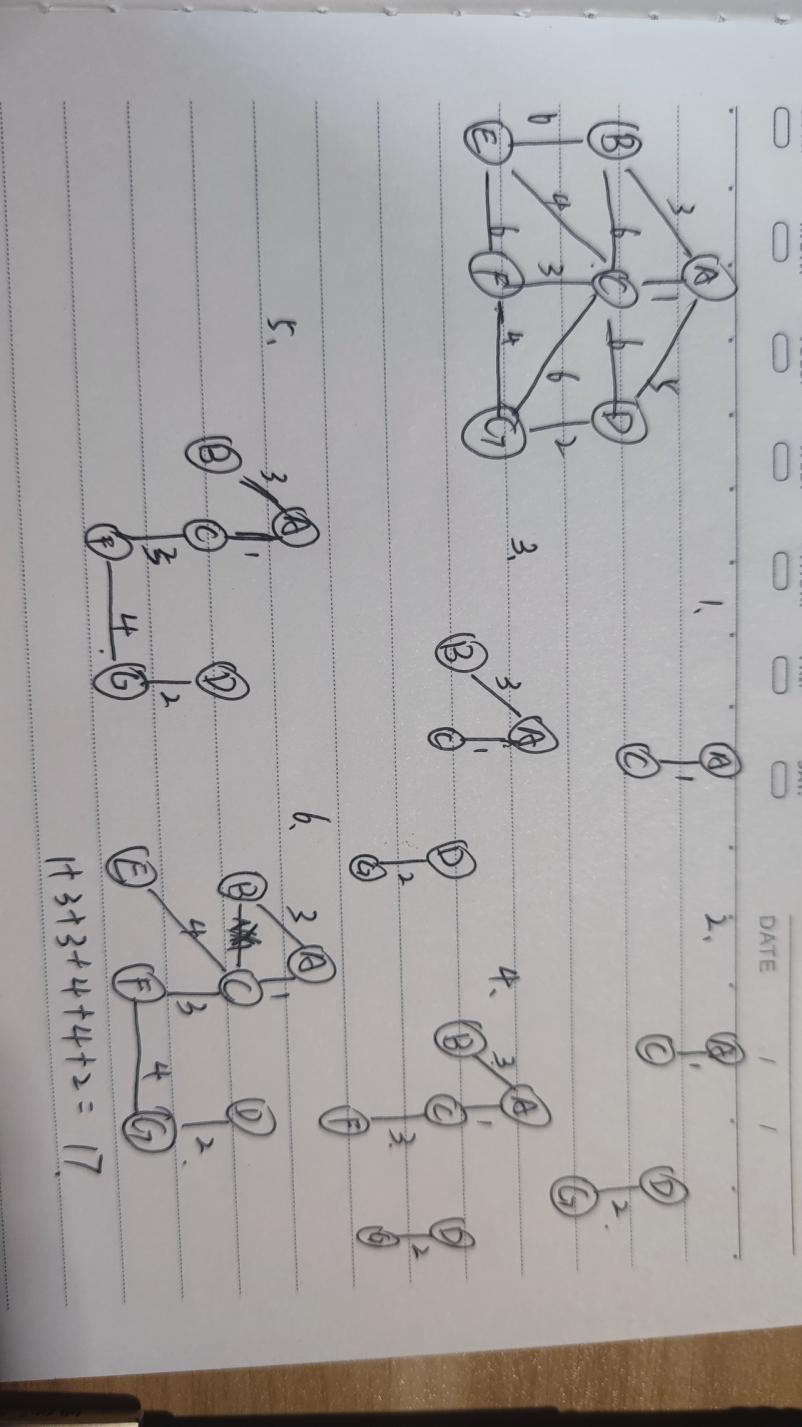
在一给定的无向图G = (V, E) 中，(u, v) 代表连接顶点 u 与顶点 v 的边（即），而 w(u, v) 代表此边的权重，若存在 T 为 E 的子集（即）且为无循环图，使得的 w(T) 最小，则此 T 为 G 的**最小生成树**。

96dda144ad345982caec25cf0df431adcbef840d

如下图样例，具有7个顶点，12条边

# 解析

[问题的理解和推导，可用电子版直接在此编写，也可用纸笔推导，拍照嵌入本文档]



# 设计

[核心伪代码]

sort(edge + 1, edge + 1 + m, cmp);//按权值排序

for (int i = 1; i <= n; i++)

fat[i] = i;//并查集初始化

for (int i = 1; i <= m; i++)//从小到大遍历

{

if (k == n - 1) break;//n个点需要n-1条边连接

if (father(edge[i].from) != father(edge[i].to))//假如不在一个团体

{

unionn(edge[i].from, edge[i].to);//加入

tot += edge[i].dis;//记录边权

k++;//已连接边数+1

}

}

# 分析

[算法复杂度推导]

# 源码

[github源码地址]

https://github.com/whx116/lab1-2.git