**实验报告**

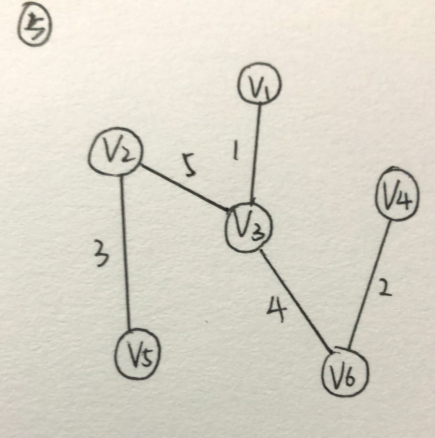
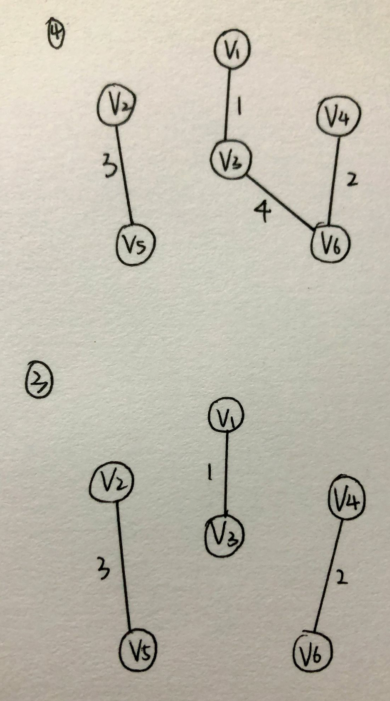
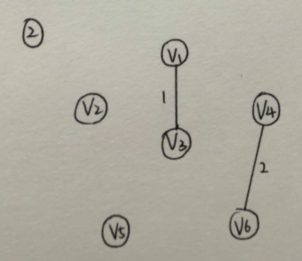
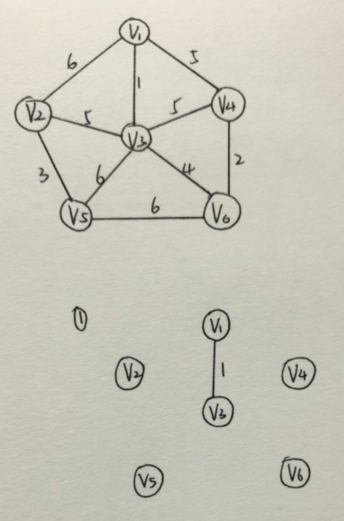
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | 《算法分析与设计》 | **实验日期** | 年 月 日 至 年 月 日 | | | | |
| **学生姓名** | 陶玮 | **所在班级** | 计算机184 | | | **学号** | 2018212212019 |
| **实验名称** |  | | | | | | |
| **实验地点** |  | | | **同组人员** | 无 | | |

# 问题

有一张无向图，该图有n个顶点，至少n-1条边。每条边都有相应的权重，计算经过所有点的权重相加最小的路径。

# 解析

从边出发，不断寻找当前未添加进Et的、且权值最小的边，若添加后不形成环，则添加成功；否则跳过，继续尝试添加下一条边。最后，判断边的数量arcnum是否是点的数量vexnum-1，若是则最小生成树构造成功，否则失败。



# 设计

int Find(int x)

{

if (father[x] == -1) return x;

return father[x] = Find(father[x]);

}

bool Union(int x, int y)

{

x = Find(x);

y = Find(y);

if (x == y) return false;

if (x > y) father[x] = y;

if (x < y) father[y] = x;

return true;

}

int cmp(graph x, graph y)

{

if (x.cost < y.cost) return true;

return false;

}

# 分析

Kruskal算法与边相关，时间复杂度为O（|E|log|E|），适合边少顶点多的图；

# 源码