

# 数据结构及其算法第五次实验

数据结构及其算法第五次实验

Content

Requirement

Input

Output

## Content

yy 最近迷上了一款叫做缺氧的游戏，他建立了一个氧气循环系统，由一台制氧机产生氧气，运输到复制人的房间，每个复制人的房间位置不同。因此所需管道的长度也不同，同时由于要保持适宜的温度，在一些区域需要建造隔热管道或导热管道，因此，到不同复制人房间所需要的管道费用不尽相同。

## Requirement

在可能假设的  $m$  条管道中，选取  $n - 1$  条管道，使得既能连通  $n$  个房间，又能使总开销最小。每条管道的费用以边的权值给出，采用 **邻接表** 的形式存储。

我们提供的接口：

```
class Graph {
    vector<vector<pair<int, float>>> adj;    // 读取时存入的数据
    vector<vector<pair<int, float>>> mst;    // 最小生成树，无向
    vector<vector<pair<int, float>>> res;    // result, 你需要注意
    size_t root;    // 答案的根节点
public:
    Graph(fstream &fin);    // 构造函数, TODO
    void writeGraph(fstream &fout);
    void saveGraph(const string &filename);
    void prim();    // TODO
    void kruskal();    // TODO
    void process();    // TODO
};
```

你需要完成 `Graph()` , `prim()` , `kruskal()` , `process()` 函数。

执行步骤：

1. `prim()` 或 `kruskal()` 函数，将生成的无向图存储到 `mst` 中。
2. 调用 `process()` 函数，将 `mst` 的一些边去除，从而形成一个有向图，这个有向图是一棵树，以 `root` 为根结点。
3. 调用 `saveGraph()` 函数

## Input

第一行 3 个数据： `n m root`， `n` 和 `m` 分别表示结点数和边数，结点的编号分别为： $0, 1, 2, \dots, n-1$ ， `root` 表示最后的结果的根节点，即上述的氧气产生设备。

而后  $m$  行： `C1 C2 weight`，表示从 `C1` 到 `C2` 的一条无向边，权值为 `weight`

## Output

由于只有一个氧气产生设备，因此最终我们得到的结果是一棵树，以氧气产生设备为 `root`，采用 `vector<vector<pair<int, float>>>` 存储，我们封装了一个函数 `saveGraph()`。用于将输出图输出到 html 文件以便查看。

