4-12：

> 试设计一个构造图G生成树的算法，使得构造出的生成树的边的最大权值达到最小。

题目分析

乍一看，以为本题与 $Kruskal$ 算法构造最小生成树如出一辙，但事实真是如此吗？

由 $Kruskal$ 算法构造最小生成树的过程如下（假设只存在一棵最小生成树）：

- 将边权从小到大排序，每次选取最小的，若生成环，则舍弃；否则保留，由 $Kruskal$ 算法的证明可知此边一定在最小生成树内。

如果本题目不能用 $Kruskal$ 算法求解，那问题就一定出现在“舍弃”的边上，如果选取舍弃的边，而不选择边权更小的边，有没有可能导致最终的生成树的最大边权更小？下面给出证明：

设 $S_k=\{e_1,e_2,\cdots,e_k\}$ 是由 $Kruskal$ 构造的最小生成树，$T_k=\{e'_1,e'_2,\cdots,e'_k\}$ 是本问题的最优解（最大边权最小）。我们假设 S_k,T_k 中的边权按下标从小到大递增，则由最优解的性质可知：$e'_k.w<e_k.w$（注：$e_i.w$表示第 i 条边的权值），而由 $Kruskal$ 算法可知，e'_k 被其舍弃的原因一定是由于形成了环路。我们假设这个环只涉及三个结点 n_1,n_2,n_3，e'_k 连接点 n_1,n_2，因此 n_1,n_3 和 n_2,n_3 之间的边必然已被选取，且这两条边的权值均小于 $e'_k.w$。而最小生成树中包含边 e_k，我们假设该边涉及的结点为：n_4,n_5，则 $Kruskal$ 选取 e_k 这条边有如下两种可能：

- n_4 已经位于最小生成树中，n_5 尚未位于最小生成树中，则连接该边的目的是为了将 n_5 加入树中。而不选取 n_5 的其他边，是因为其他边的边权更大，e_k 是连接 n_5 结点的所有边中边权最小的，但又因为 $e'_k.w<e_k.w$，说明存在一条更小的边连接 n_5，矛盾。
- n_5 已经位于最小生成树中，n_4 尚未位于最小生成树中。与上述结论完全一致，必然能推出矛盾。

因此，由上述证明可知，本题的最优解其实就是 $Kruskal$ 所构造出的最小生成树，该树同时保证了最大边权最小。

代码实现

代码实现详见4-12 Code.cpp

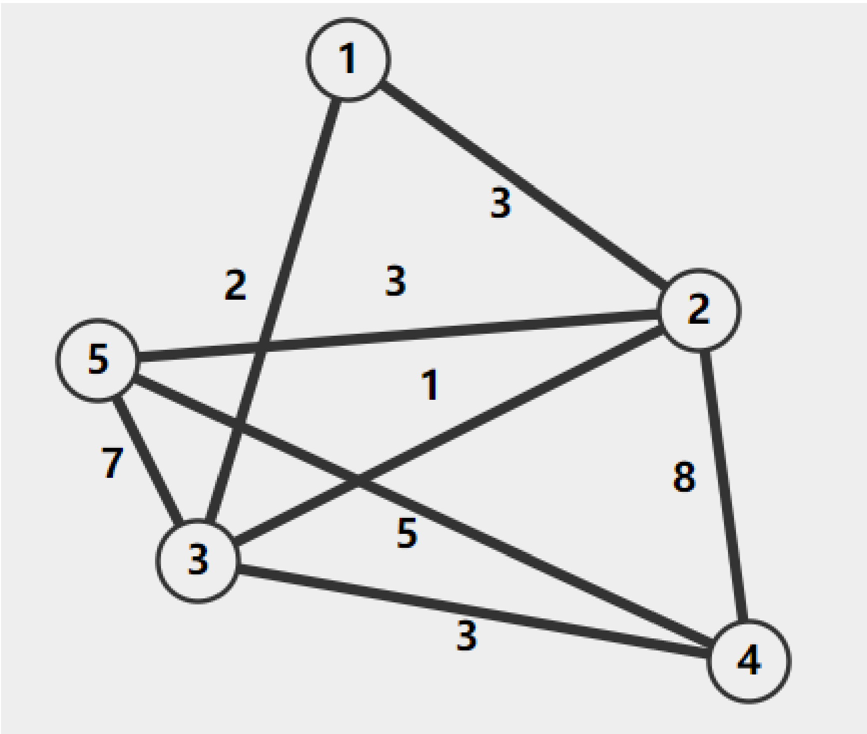
输出示例

规定输入格式如下，首先输入一行：

- 第一行两个正整数 n,m，n 表示结点个数（结点从 1 开始编号），m 表示边的个数。
- 接下来 m 行，每行输入三个数 x,y,z，分别表示起始结点，终止结点和边权。

```
输入：
```

```
5 8
1 2 3
1 3 2
2 3 1
3 5 7
4 5 5
2 4 8
3 4 3
2 5 3
输出：
2 3 1
1 3 2
3 4 3
2 5 3
```



样例分析：对于本样例，Kruskal给出的是1-3、2-5、3-4、2-3四条边，显然，这也是最大边权最小的图。

算法分析

本算法所涉及的部分如下：

- 输入输出：需要 $O(E)$ 复杂度。
- 预处理：将边塞入最小堆中需要 $O(ElogE)$ 复杂度。
- $Kruskal$：通过并查集选取 $n-1$ 条边：需要 $O(V+E)\alpha(V)$ 复杂度，但要比 $O(ElogE)$ 小。

综上，本算法的复杂度为 $O(ElogE)$。