

Kruskal算法实现



主讲人：邓哲也



Kruskal算法执行过程

Kruskal 算法在每选择一条边加入到生成树集合 T 时，有两个关键步骤如下：

1. 从 E 中选择当前权值最小的边 (u, v)

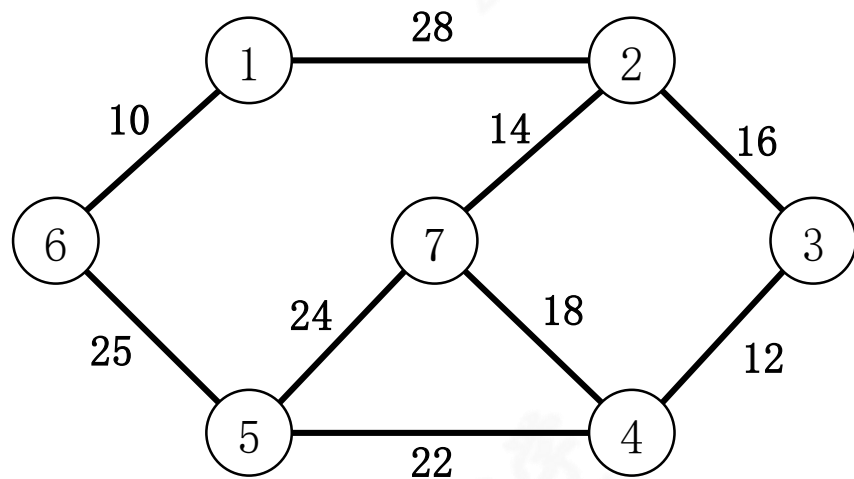
用最小堆 或者 按照边权大小排序实现

2. 选择权值最小的边后，要判断两个顶点是否属于同一个连通分量，如果是，要舍去；如果不是，要合并它们。

用并查集实现

Kruskal算法执行过程

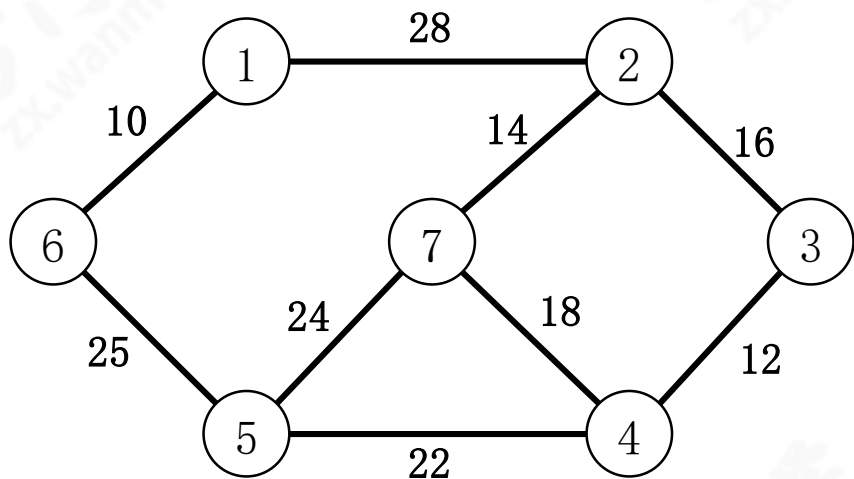
以下面这个无向图为例，我们来看看 Kruskal 算法执行过程。



Kruskal算法执行过程

初始状态:

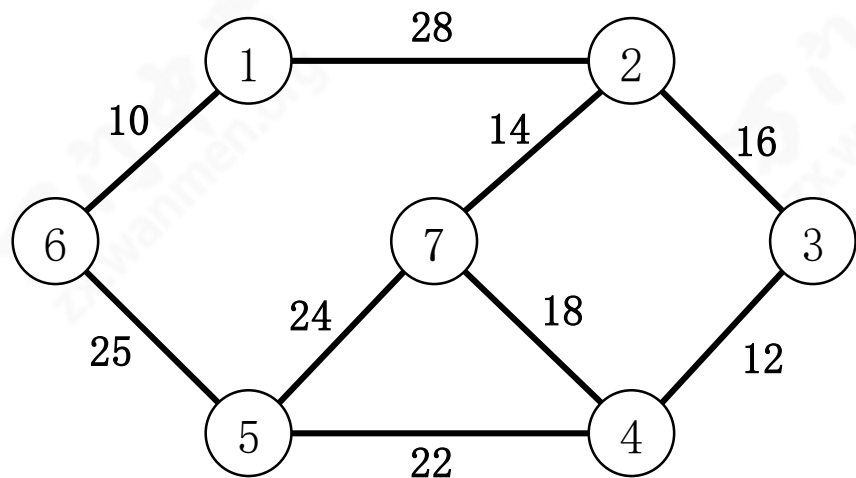
并查集的初始状态为各个顶点各自构成一个连通分量。



	1	2	3	4	5	6	7
parent[]	1	2	3	4	5	6	7

Kruskal算法执行过程

9条边已经从小到大排好序了。

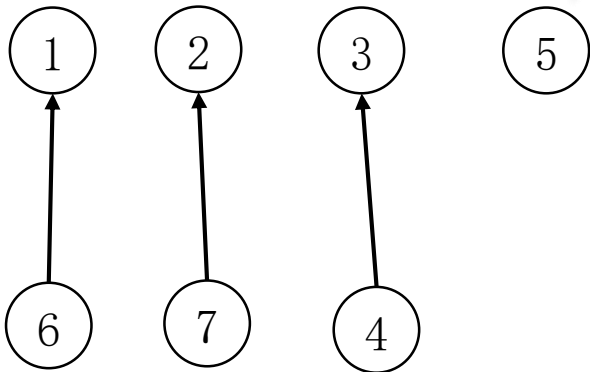
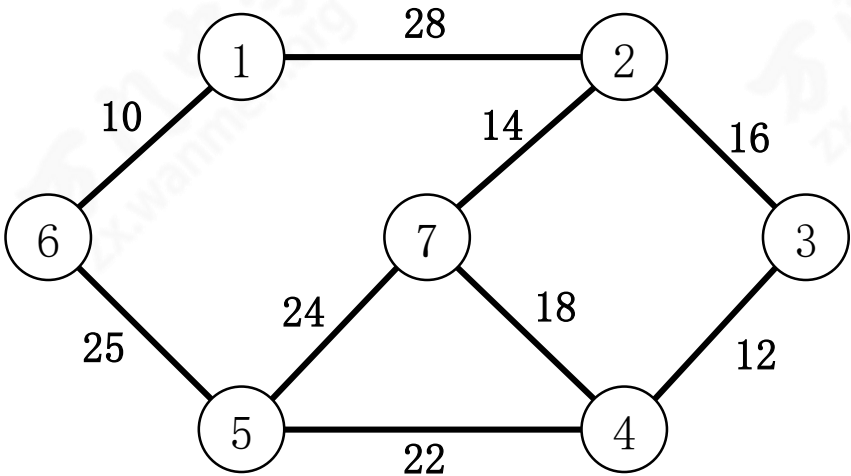


	1	2	3	4	5	6	7
parent[]	1	2	3	4	5	6	7

u	v	w
1	6	10
3	4	12
2	7	14
2	3	16
4	7	18
4	5	22
5	7	24
5	6	25
1	2	28

Kruskal算法执行过程

选择(1, 6), (3, 4), (2, 7)三条边后, 状态更新:

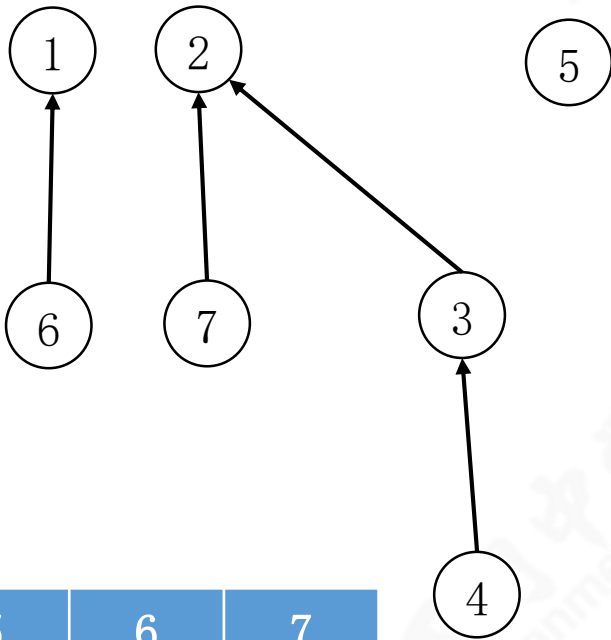
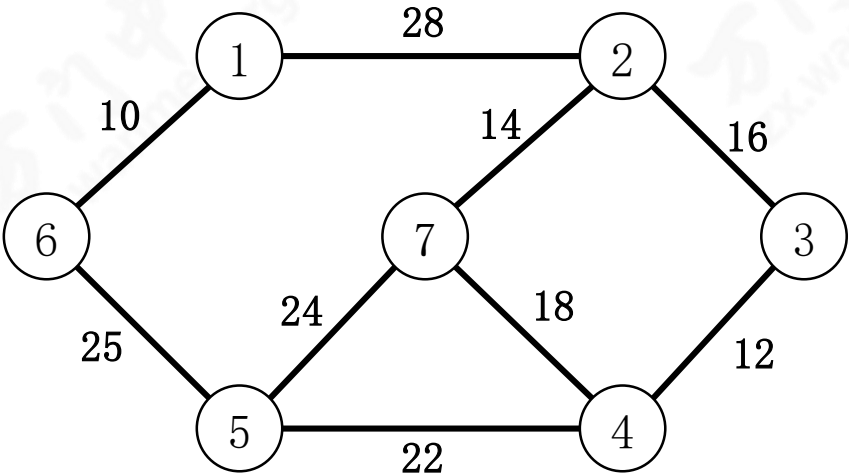


	1	2	3	4	5	6	7
parent[]	1	2	3	3	5	1	2

u	v	w
1	6	10
3	4	12
2	7	14
2	3	16
4	7	18
4	5	22
5	7	24
5	6	25
1	2	28

Kruskal算法执行过程

选择 (2, 3) 后，需要合并2和3所在的连通分量，状态更新：



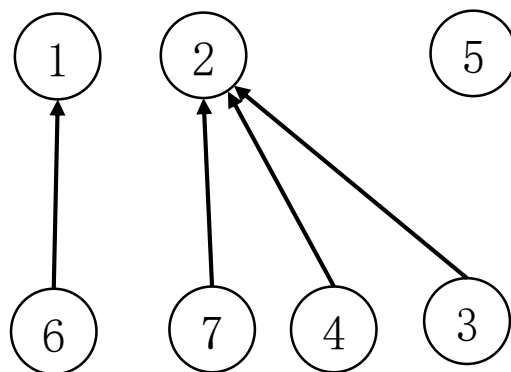
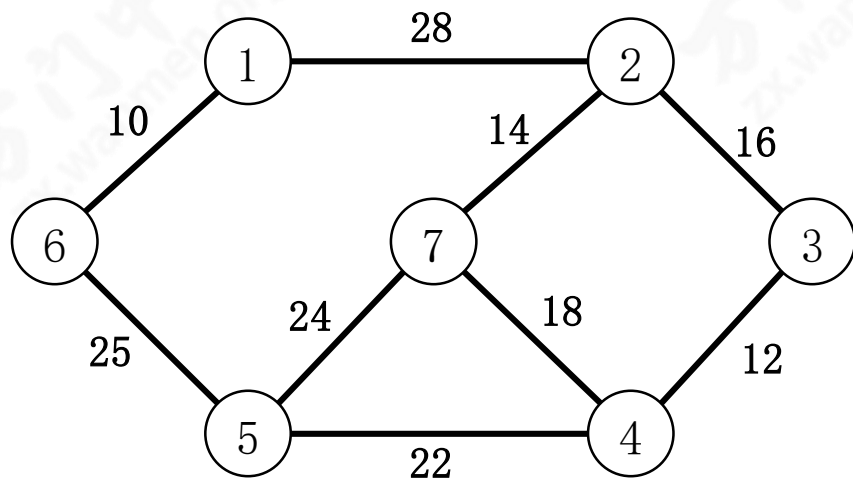
	1	2	3	4	5	6	7
parent[]	1	2	2	3	5	1	2

u	v	w
1	6	10
3	4	12
2	7	14
2	3	16
4	7	18
4	5	22
5	7	24
5	6	25
1	2	28

Kruskal算法执行过程

选择(4, 7)时，因为两个点位于同一个连通分量，这条边会被弃用。

但在查找4的根节点时，会压缩路径，更新状态：

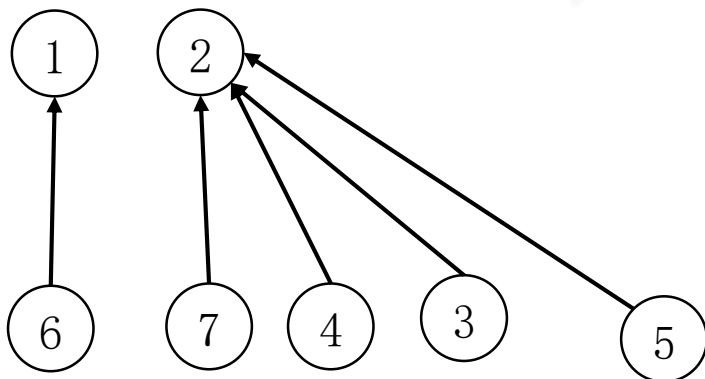
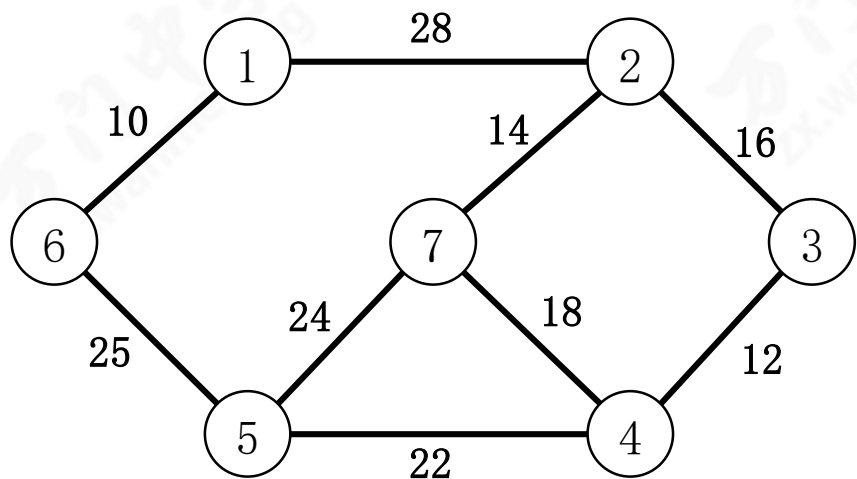


	1	2	3	4	5	6	7
parent[]	1	2	2	2	5	1	2

u	v	w
1	6	10
3	4	12
2	7	14
2	3	16
4	7	18
4	5	22
5	7	24
5	6	25
1	2	28

Kruskal算法执行过程

选择(4, 5)时, 根据按秩合并的规则, 5的根指向4的根。

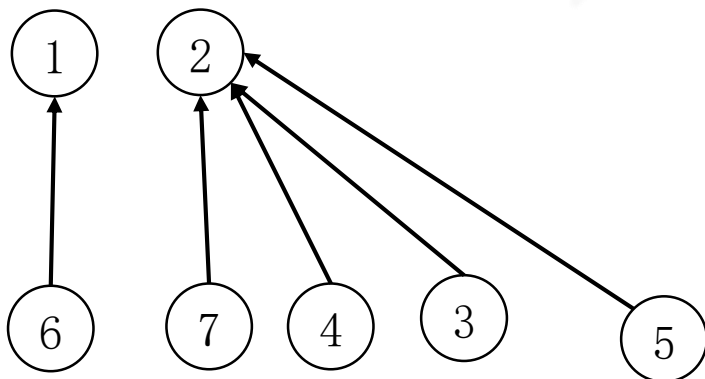
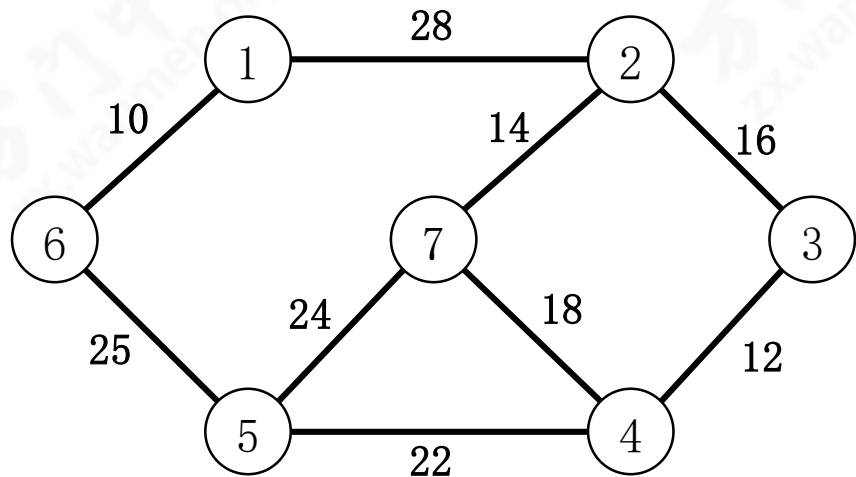


	1	2	3	4	5	6	7
parent[]	1	2	2	2	2	1	2

u	v	w
1	6	10
3	4	12
2	7	14
2	3	16
4	7	18
4	5	22
5	7	24
5	6	25
1	2	28

Kruskal算法执行过程

选择 (5, 7) 时, 因为两个点位于同一个连通分量, 这条边会被弃用。

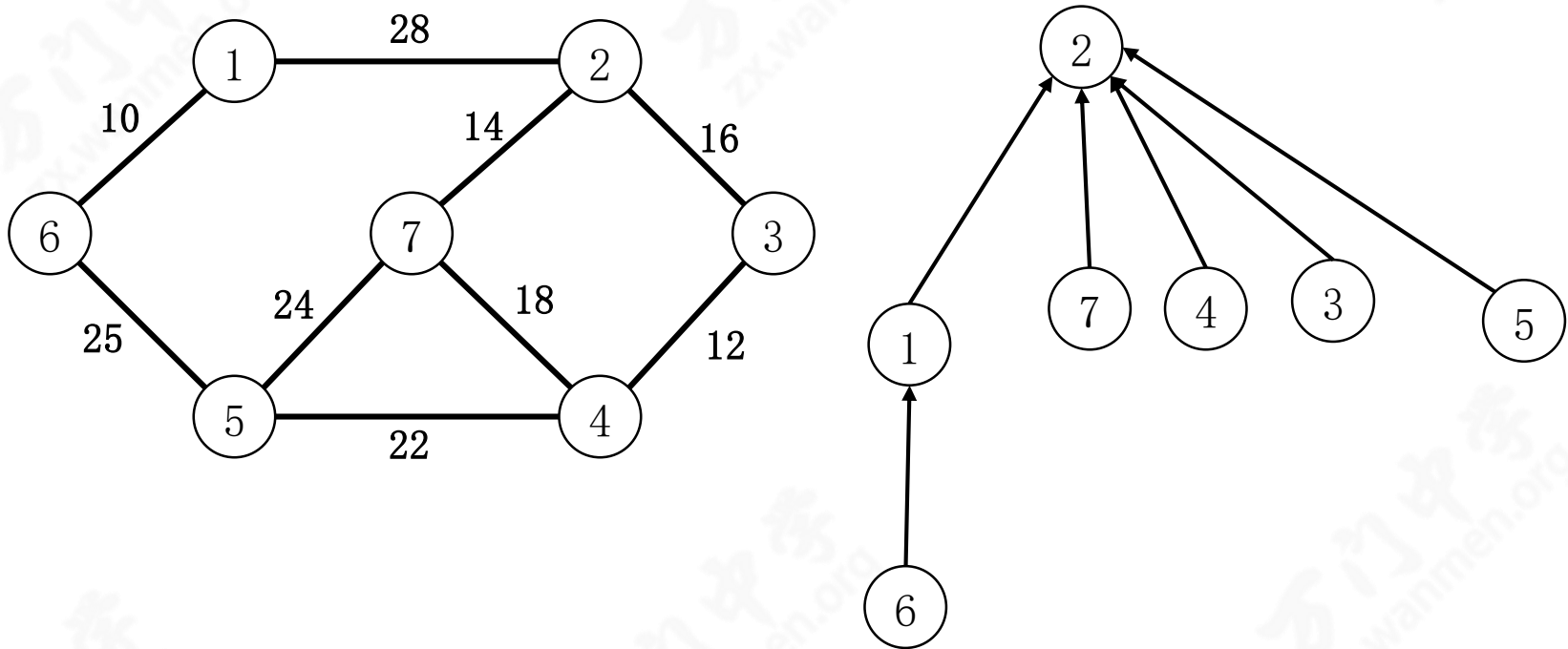


	1	2	3	4	5	6	7
parent[]	1	2	2	2	2	1	2

u	v	w
1	6	10
3	4	12
2	7	14
2	3	16
4	7	18
4	5	22
5	7	24
5	6	25
1	2	28

Kruskal算法执行过程

选择 (5, 6) 时, 根据按秩合并的规则, 6 的根会指向 5 的根。
选出了 $n-1$ 条边, 算法结束。



	1	2	3	4	5	6	7
parent[]	2	2	2	2	2	1	2

u	v	w
1	6	10
3	4	12
2	7	14
2	3	16
4	7	18
4	5	22
5	7	24
5	6	25
1	2	28

下节课再见