

知识精炼（二）



主讲人：邓哲也

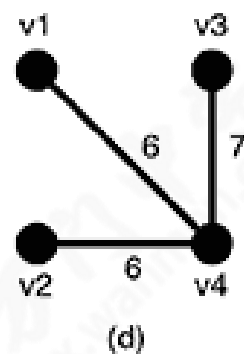
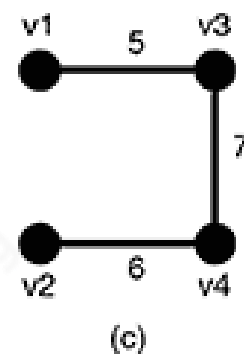
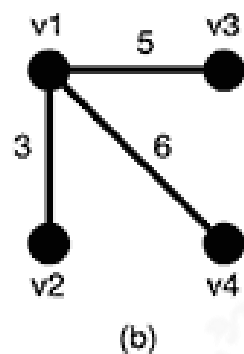
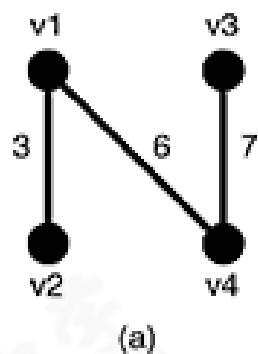
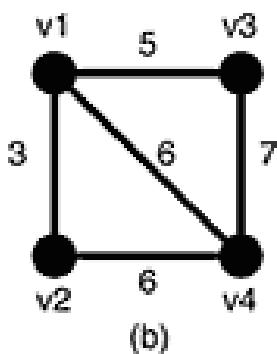
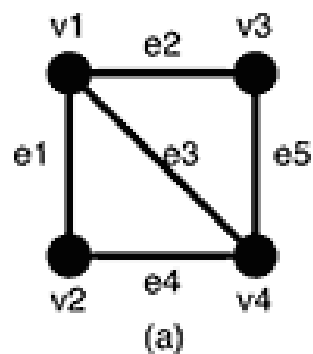


P0J3522 Slim Span

给定一张无向图，你可以找到很多个生成树。

现在希望你能找出一个生成树，使得生成树中“最大边-最小边”的值最小。

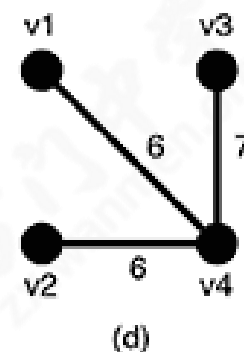
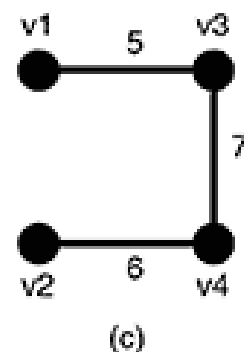
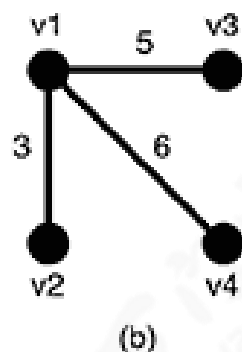
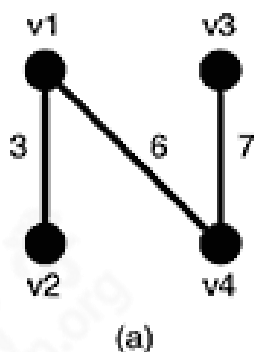
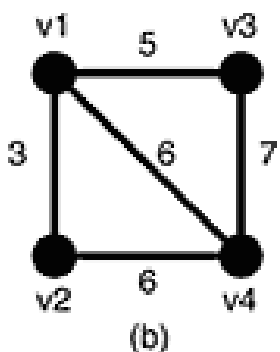
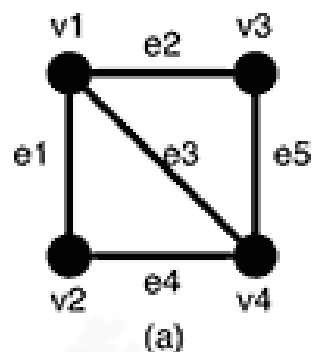
$n \leq 100$



P0J3522 Slim Span

我们不妨考虑，枚举生成树中的“最小边” e 。

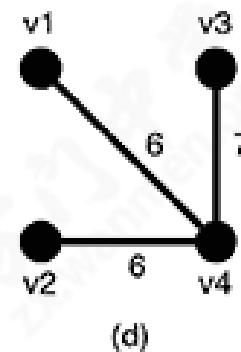
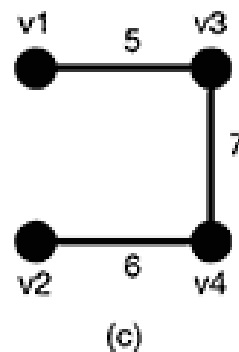
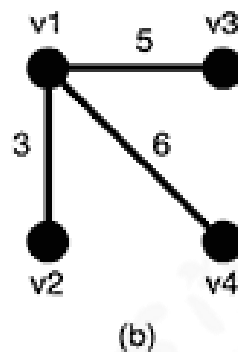
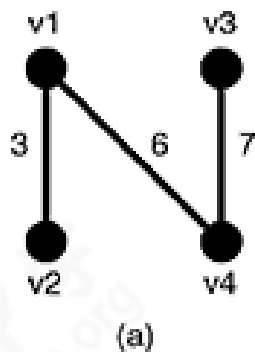
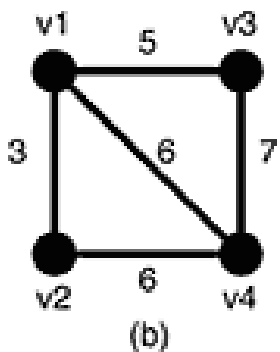
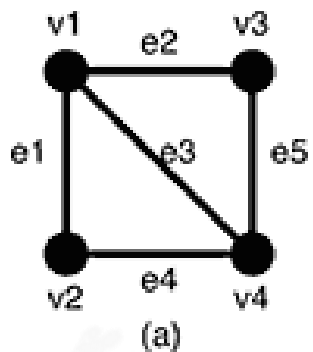
接下来只要删除所有边权小于 e 的边，然后在剩下的图上跑Kruskal算法，得到最小生成树之后，找出“最大边”，用“最大边” - “最小边”去更新答案。



P0J3522 Slim Span

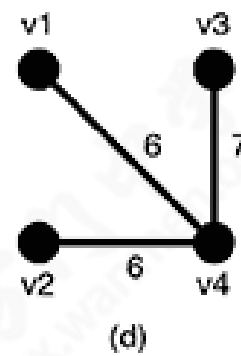
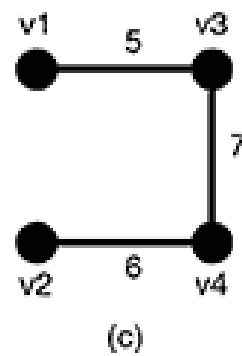
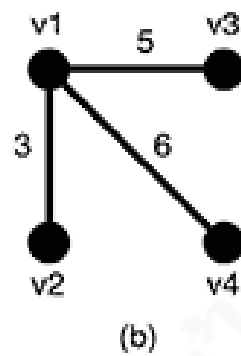
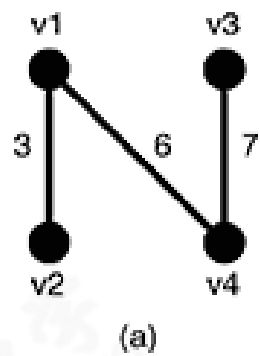
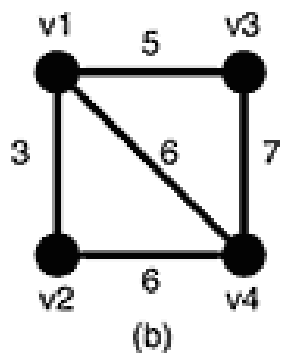
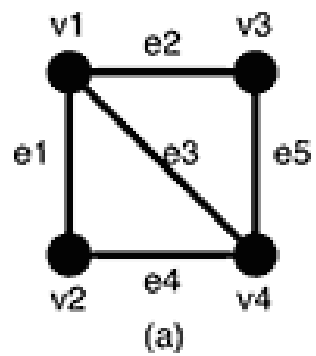
正确性?

如果存在某个生成树的“最大边”比最小生成树中的“最大边”还小呢?



P0J3522 Slim Span

反证法，不存在。

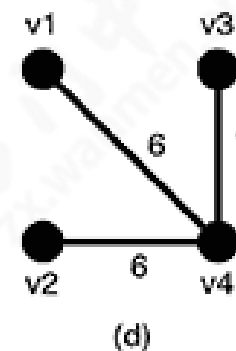
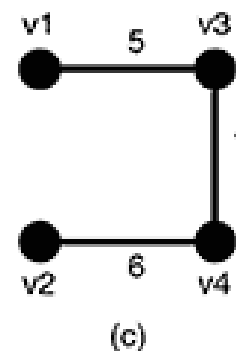
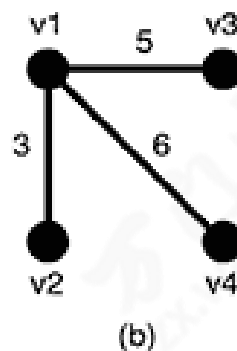
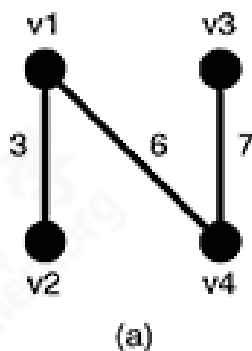
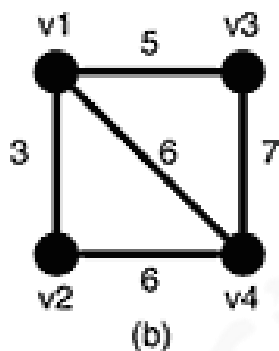
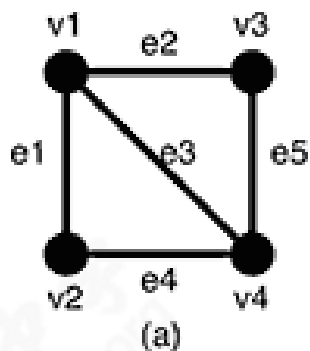


P0J3522 Slim Span

因此我们的想法是正确的。

每次枚举最小边，然后把大于等于最小边权值的边保留，按照边权从大到小排序，跑 Kruskal 算法。

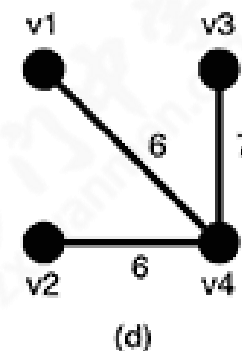
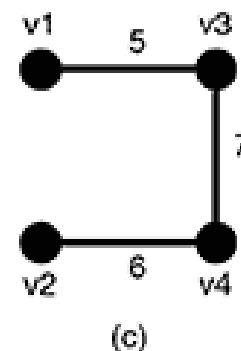
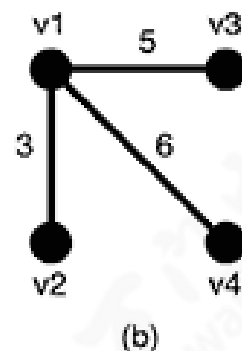
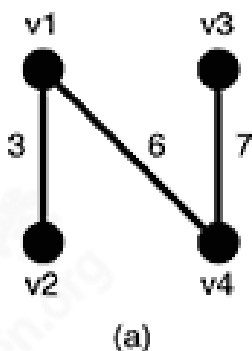
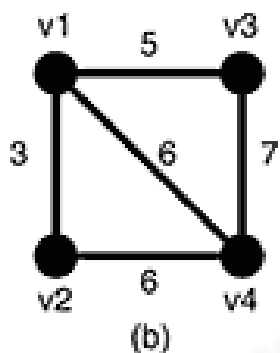
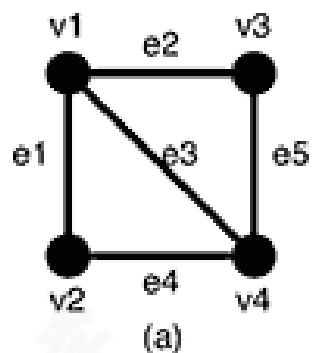
因为边的数量是 $O(n^2)$ 级别的，做一次Kruskal算法需要 $O(n^2 \log n)$ 的时间复杂度。总时间复杂度为 $O(n^4 \log n)$ 。很悬。



P0J3522 Slim Span

很简单。只要一开始对所有边排一次序就可以了。

总时间复杂度为 $O(n^4)$ 。完美解决。



下节课再见