23.1-5

反证法：假设图 G 最小生成树一定都包含权重最大的边 e

证明：对于边 e 所在的环路 v1 -e- v2 -……-v_{n-1} -e'-v1，w(e)>w(e')，对于最小生成树 T，若删去边 e，加入边 e'，树中仍然不会有该环路，新的生成树 T'的权重小于原来的树 T 的权重，说明 T 并不是最小生成树，与假设矛盾

所以，图 G 存在一颗不包含边 e 的最小生成树

23.2-1

假设想选择 T 作为最小生成树。然后，为了使用 Kruskal 算法获得此树，我们将首先按边的权重对边进行排序, 然后通过选取包含在最小生成树中的一条边来解决边权重中的连接，并将所有不在 T 中的边权视为稍大，即使它们具有相同的实际权重。 通过这种排序，我们能够找到与所有最小生成树 w(T)具有相同权重的树。但是，由于我们优先考虑在 T 中的边，因此我们将选择 T 中的边，而不是其他最小生成树中可能存在的任何其他边。

23.2-8

反例: 四个顶点 a、b、c 和 d。设边分别为 （a， b）、（b、c）、（c、d）、（d、a），权重分别为 1、5、1 和 5。 V1 = {a， d} 和 V2 = {b， c}。然后每个树上只有一个边缘入射，因此在 V1 和 V2 上的最小生成树分别由边缘 （a， d） 和 （b， c） 组成，总权重为 10。加上连接它们的权重 1 边缘，我们得到权重 11。但是，MST 将使用两个权重 1 边和仅使用权重 5 边中的一个，总权重为 7。

23-3

a . 最小生成树是每次从图中加入边权最小的安全边，e1e2……en，其中最后加入的边 en 则是树中最大边权的边，若边 en 比 G 中其他生成树中的最大边权重大，说明有其他比 en 小的边 e'可以加入最小生成树，那么它就会先 en 之前加入到树中，e'才是最小生成树中边权最大的边，比其他生成树的最大边都小，所以最小生成树是瓶颈生成树

b . （1）边排序：将所有权重不超过 b 的边按权重升序排序。

（2）构建生成树：按顺序每加入一条边(u,v)，如果它们位于不同的连通分量中，则合并这两个连通分量，并将这条边加入到生成树中。如果最终能生成一个生成树，所有顶点都在树中，则说明存在一个瓶颈生成树，其最大边权不超过 b。

c. （1）筛选边：把边权超过 b 的删除，按权重升序排序

（2）构建生成树：按顺序每加入一条边(u,v)，如果它们位于不同的连通分量中，则合并这两个连通分量，并将这条边加入到生成树中。如果最终能生成一个生成树，所有顶点都在树中，则说明存在一个瓶颈生成树，其最大边权不超过 b。

23-4

a. 不正确，这个算法按照边的权重降序排列，每次都加入最大的边，最后的结果是最大生成树

b. 不正确，随机的放入边，只保证没有环，会错过一些边权更小的边

c. 正确，随机放入边，一旦出现环就删除边权最大的边，最后留下的就是最小生成树