hw10 due 11.25

> 21.2-3, 21.3-2；22.2-5，22.3-8；23.1-3，23.2-8；

21.2-3

21.2-3 对定理 21.1 的整体证明进行改造，得到使用链表表示和加权合并启发式策略下的 MAKE-SET 和 FIND-SET 的摊还时间上界为 $O(1)$，以及 UNION 的摊还时间上界为 $O(\lg n)$。

在证明定理21.1时，我们得出结论，n次UNION操作的运行时间最多为$O(n\lg(n))$。这意味着它们每一个的平摊时间最多为$O(\lg(n))$。同时，由于MAKE-SET和FIND-SET操作执行的实际工作量只有一个常数级别，并且这种简便性并未用于抵消UNION操作的成本，所以它们都具有$O(1)$的运行时间。

21.3-2

21.3-2 写出使用路径压缩的 FIND-SET 过程的非递归版本。

为了非递归地实现FIND-SET，设x为我们调用函数的元素。创建一个链表A，其中包含一个指向x的指针。每次我们在树中向上移动一个元素时，将该元素的指针插入到A中。一旦找到根节点r，就使用链表找到从根到x路径上的每个节点，并更新它们的父节点为r。

22.2-5

22.2-5 证明：在广度优先搜索算法里，赋给结点 u 的 $u.d$ 值与结点在邻接链表里出现的次序无关。使用图 22-3 作为例子，证明：BFS 所计算出的广度优先树可以因邻接链表中的次序不同而不同。

首先要证明的是，分配给顶点的值d与邻接表中条目的顺序无关。这一点依赖于定理22.5的证明，该定理验证了广度优先搜索（BFS）的正确性。特别地，存在$\nu.d=\delta(s,\nu)$在过程结束时。因为$\delta(s,\nu)$是底层图的固有属性，无论选择何种邻接表形式表示图，这个值都不会改变。因此，由于d值等同于在变更邻接表时不会改变的属性，修改邻接表时，它也保持不变。

接下来，为了证明π确实取决于邻接表的排序，可以参考图22.3。注意，在给定的过程中，在w的邻接表里，t出现在x之前。同时，在该过程中，有$u.\pi=t$。现在，假设在w的邻接表中x排在t之前，x将会比t更早地被加入队列，意味着在处理t的子节点之前，x已经成为u的子节点。这样一来，在w的邻接表中不同的排序会导致$u.\pi=x$。

22.3-8

22.3-8 请给出如下猜想的一个反例：如果有向图 G 包含一条从结点 u 到结点 v 的路径，并且在对图 G 进行深度优先搜索时有 $u.d<v.d$，则结点 v 是结点 u 在深度优先森林中的一个后代。

考虑一个包含三个顶点 u, v, 和 w 的图，以及边 $(w,u),(u,w)$ 和 (w,v)。假设深度优先搜索（DFS）首先探索顶点 w，且 w 的邻接表中 u 排在 v 之前。接下来，搜索发现顶点 u。u 的唯一邻接顶点是 w，但 w 已经是灰

色的，所以 u 的搜索完成。由于顶点 v 此时尚未成为 u 的后代，且 u 已经搜索完成，所以 v 不可能成为 u 的后代。

23.1-3

> **23.1-3 证明：如果图 G 的一条边 (u, v) 包含在图 G 的某棵最小生成树中，则该条边是横跨图 G 的某个切割的一条轻量级边。**

设 T_0 和 T_1 是通过从最小生成树（MST）中移除边 (u, v) 获得的两棵树。假设 V_0 和 V_1 分别是 T_0 和 T_1 的顶点集。考虑将 V_0 与 V_1 分开的切分。假设与此相矛盾的是，在这个切分中存在某条边的权重小于 (u, v) 的权重。那么，可以通过将该边添加到 $T_1 \cup T_0$ 中，构造出整个图的一个最小生成树。这将导致一个比包含 (u, v) 的原始最小生成树权重更小的最小生成树。

23.2-8

> **23.2-8 Borden 教授提出了一个新的分治算法来计算最小生成树。该算法的原理如下：给定图 $G=(V, E)$，将 V 划分为两个集合 V_1 和 V_2，使得 $|V_1|$ 和 $|V_2|$ 的差最多为 1。设 E_1 为端点全部在 V_1 中的边的集合，E_2 为端点全部在 V_2 中的边的集合。我们递归地解决两个子图 $G_1=(V_1, E_1)$ 和 $G_2=(V_2, E_2)$ 的最小生成树问题。最后，在边集合 E 中选择横跨切割 V_1 和 V_2 的最小权重的边来将求出的两棵最小生成树连接起来，从而形成一棵最后的最小生成树。**
> **请证明该算法能正确计算出一棵最小生成树，或者举出反例来明说该算法不正确。**

教授Borden的观点存在错误。考虑一个包含4个顶点的图：a, b, c 和 d。设边为$(a, b), (b, c), (c, d), (d, a)$ ，它们的权重分别为1, 5, 1, 和 5。设定 $V_1 = a, d$ 和 $V_2 = b, c$。那么，每个集合上只有一条边，所以必须在V_1 和 V_2 上取的树分别由边 (a, d) 和 (b, c) 构成，总权重为10。加上连接这两部分的权重为1的边，总权重达到11。然而，一个最小生成树（MST）将使用两条权重为1的边和一条权重为5的边，总权重只有7。