

数据结构与算法 I 作业 22

2019201409 于倬浩

2020 年 12 月 23 日

21.2-5

将每个元素中用来存储指向集合对象的指针指向链表的末尾即可。对于链表末尾的元素，将这个指针指向集合对象。这样，原始状态下从每个元素跳到链表末端需要先跳到集合对象，再跳到链表末端，更改后每个元素可以直接一次跳到链表末端，然后第二次跳到集合对象，不再需要集合对象中维护指向链表末端的指针。

合并时，只需将较小集合放在前面，更改每个元素指向末尾的指针到较大集合的末尾元素，最后将较小集合的末尾元素的链表指针改为较大集合的表头即可，运行时间依旧是关于较小集合元素个数的线性函数。

21.3-3

为了便于分析，假设 m 大于 $2n$ 。调用 n 次 **Make-Set** 新建 n 个单元素集合。接下来，使用 $n-1$ 次 **Union**，每次将元素数目最小的两个集合合并起来，最终得到一棵深度为 $\Theta(\lg n)$ 的树。对于剩下的操作，任意找一个叶子节点不断进行 **Find-Set**，这样每次 **Find-Set** 的运行时间均为 $\Theta(\lg n)$ ，这也是一次 **Find-Set** 操作的运行时间上界。这样渐进意义上，如果 m 远大于 n ，可以确定运行时间的上界为 $\Omega(m \lg n)$ 。

对于 $m \leq n$ 的情况，或是 m 与 n 数量级接近的情况，只需要减少 **Union** 的次数，例如运行 $\frac{m}{2}$ 次合并，得到若干棵深度为 $O(\lg n)$ 的树，随便找一个叶子节点不断 **Find-Set**，依旧可以确定运行时间上界为 $\Omega(m \lg n)$ 。