作业8解答：

1. 设$S_1, S_2, \cdots, S_k$是k个整数的集合，其中的整数值域从1到n，这k个集合的元素个数之和为n。设计一个$O(n)$时间的算法，将所有S_i排序$(i=1,2,\cdots,k)$。

解：设每个集合S_j是一个链表，链表的表元是S_j中的元素。

设A是n个元素的数组，每个数组元素$A[i]$是一个链表，链表表元是集合元素i所属的集合S_j $(1\le j\le k)$的名字。一个元素i可能属于若干集合，故$A[i]$链表可能有若干个表元。

算法如下：

```
FOR i:=1 TO n DO
   A[i]:=NIL;
FOR j:=1 TO k DO
   FOR 每个i属于Sj DO
   BEGIN
      从Sj中取出元素i;
      将j加入链表A[i];
   END;
/*做到这里，每个Sj为空*/.
FOR i:=1 TO n DO
   IF A[i]≠NIL THEN
      FOR 链表A[i]中的每个表元j DO
      BEGIN
         将j从A[i]取出;
         将i加到链表Sj末尾;
      END;
```

算法分析：

本算法扫描每个集合 S_i 的每个元素一遍，因为所有 S_i ($i=1,2,\cdots,k$) 共有 n 个元素，所以这一步需 $O(n)$ 时间。然后扫描 A 中集合元素 1 到 n，$A[i]$ 中的链表元为集合元素 i 所属的集合，共有 n 个集合元素，故 $A[1], A[2], \cdots, A[n]$ 中共有 n 个链表表元。故第 2 步也是 $O(n)$ 时间。整个算法时间复杂度为 $O(n)$。

2. 证明快速排序在最好情况下运行时间为 $\Omega(n\log_2 n)$。

证明：由于快速排序总的运行时间等于所有递归调用中要排序的元素个数之和，也等于每个元素出现在不同递归调用中的次数之和，即每个元素排好序时所在的层数之和。快速排序算法最好情况下，每次划分 (Partition) 将一组元素 $A[i], A[i+1], \cdots, A[j]$ 划分成元素一样多的两组，所有元素 r_i 所在的层数都是最后一层 ($k+1$ 层)，第 $k+1$ 层有 $n=2^k$ 个叶子，而该树是一个满二叉树，$k=\log_2 n$。

各元素的层数之和为 $n(k+1)=n(\log_2 n+1)$。

故最好情况下时间复杂度为 $\Omega(n\log_2 n)$。

装 订 线