CS 610, Spring 2009, Prof. Calvin
Homework #3 Solutions

C-3.8 Starting with an arbitrary n node binary tree we can convert it into a left chain as follows.

(1) Let x be the root of the tree.
(2) If x has a right child, perform a rotation moving it to the location occupied by x (the inverse of the rotation shown in Figure 3.14 (b) on page 156 of the text). Keep repeating this step as long as x has a right child.
(3) Replace x by its left child, and if not null return to step 2.

Notice that with each rotation the length of the left chain (which was initially at least 1) increases by 1, and so $n-1$ rotations suffice.

Suppose we convert trees T_1 and T_2 to left chains using the above procedure (using at most $2n-2$ rotations). To convert T_1 to T_2 using at most $2n-2$ rotations, convert T_1 to a left chain, and then apply the inverse of the sequence used to convert T_2 to a left chain.

C-3.24 For $0 \leq q < p < n$ set $A_{q,p} = a_q + a_{q+1} + \cdots + a_p$. Following the hint, when we construct the subtree corresponding to the subset $(x_q, x_{q+1}, \ldots, x_p)$, put x_j at the root where j is the largest index with

$$\sum_{i=q}^{j-1} a_i < \frac{1}{2} A_{q,p}.$$

Then x_i will have depth d if and only if $(1/2)^{d+1}A < a_i \leq (1/2)^d A$, equivalently, if and only if

$$d \leq \lg\left(\frac{A}{a_i}\right) < d+1.$$

C-3.30 One choice is a skip list, with a linked list associated with each node corresponding to a key.

C-4.1 Return **true** the first time the comparison in the merge results in equality.

C-4.8 Let N_i denote the number of comparisons made with element A_i prior to A_i being selected as the pivot element, and let N be the random variable denoting the total number of comparisons made by randomized quicksort to sort the array $A_1, A_2, \ldots, A_n$. Clearly $\sum_{i=1}^n N_i = N$, and each N_i has the same distribution.

After each partition, A_i will be in a smaller subset (until it is finally selected as the pivot). It is likely to be in a substantially smaller subset; with probability $1/2$, the randomly chosen pivot element will be chosen from a subarray of length q between the $(q/4)$th largest and the $(3q/4)$th largest. This means that the largest subinterval resulting from the partition will be at most of size $3q/4$ (we can ignore integer roundoff). Call such a partition a "good partition". Since each good partition reduces the size of the subarray by at least a factor of $3/4$, we need at most $\log_{4/3}(n)$ good partitions before A_i becomes the pivot. Consider the event

$$\{N_i > 16 \log_{4/3}(n)\},$$

which is a subset of the event that in a sequence of $16 \log_{4/3}(n)$ partitions, there are fewer than $16 \log_{4/3}(n)$ good partitions. The probability of a good partition is

1/2, so the probability of the latter event is at most

$$2^{-2\log_{4/3}(n)} = n^{-\log_{4/3}(4)} \leq \frac{1}{n^3}$$

by the hint. Therefore,

$$P\left(N_i > 16\log_{4/3}(n)\right) \leq \frac{1}{n^3}$$

and

$$\begin{aligned} P\left(N > 16\log_{4/3}(n)\right) &\leq P\left(\cup_{i=1}^{n}\{N_i > 16\log_{4/3}(n)\}\right) \\ &\leq nP\left(N_i > 16\log_{4/3}(n)\right) \leq n\frac{1}{n^3} = \frac{1}{n^2}. \end{aligned}$$