CS 819B - Spring 2004 - Homework 3

Assigned February 17, due February 26

Both of the following problems ask for annotated specifications. These should contain enough annotations to make the verification conditions reasonably evident.

1. Write an annotated specification for a program that removes duplicate values from an ordered array segment. The program should assign only to X and c (and local variables), should require a time of order of the size of a **to** b, should not use any local arrays, and should satisfy

$$\begin{array}{c}\textbf{coninv } \mathsf{a} - 1 \leq \mathsf{b} \wedge \mathsf{a} \textbf{ to } \mathsf{b} \in \textbf{dom } \mathsf{X} \textbf{ in} \\ [\mathsf{X} = \mathsf{X}_0 \wedge \textbf{ord}_{\leq}\ (\mathsf{X}_0 \rceil \mathsf{a} \textbf{ to } \mathsf{b})] \\ \vdots \\ [\mathsf{a} - 1 \leq \mathsf{c} \leq \mathsf{b} \wedge \textbf{ord}_{<}\ (\mathsf{X} \rceil \mathsf{a} \textbf{ to } \mathsf{c}) \wedge \\ \{\mathsf{X} \rceil \mathsf{a} \textbf{ to } \mathsf{c}\} = \{\mathsf{X}_0 \rceil \mathsf{a} \textbf{ to } \mathsf{b}\}].\end{array}$$

2. In the worst case, the calling tree for quicksort can be extremely unbalanced. For example, if all elements of the

array X have the same value, then

quicksort$(1, 8, \mathsf{X})$ will call
    quicksort$(1, 7, \mathsf{X})$ and quicksort$(8, 8, \mathsf{X})$
quicksort$(1, 7, \mathsf{X})$ will call
    quicksort$(1, 6, \mathsf{X})$ and quicksort$(7, 7, \mathsf{X})$
quicksort$(1, 6, \mathsf{X})$ will call
    quicksort$(1, 5, \mathsf{X})$ and quicksort$(6, 6, \mathsf{X})$
    $\vdots$

Even though there are no local arrays, the local variable c, plus some storage space used by the procedure linkage mechanism, will be allocated at each level of recursion, so that the total storage needed will be proportional to the size of the array segment being sorted.

To overcome this problem, one can combine recursion with iteration: quicksort$(\mathsf{a}, \mathsf{b}; \mathsf{X})$ is redefined to be a **while** command whose body does "Prepare", followed by a single recursive call that sorts the smaller of the two subsegments, followed by a return to the beginning of the loop to sort the larger of the two subsegments. In the case shown above, for example, the main-level call quicksort$(1, 8, \mathsf{X})$ would directly call quicksort$(8, 8, \mathsf{X})$, quicksort$(7, 7, \mathsf{X})$, ..., and quicksort$(1, 1, \mathsf{X})$.

Give an annotated specification for this version of quicksort.