*Name*

COP 4531

Assignment 2

September 11, 2011

p. 188

***7-4 Stack depth for quicksort***

The QUICKSORT algorithm of Section 7.1 contains two recursive calls to itself. After QUICKSORT calls PARTITION, it recursively sorts the left subarray and then it recursively sorts the right subarray. The second recursive call in QUICKSORT

is not really necessary; we can avoid it by using an iterative control structure. This technique, called ***tail recursion***, is provided automatically by good compilers. Consider the following version of quicksort, which simulates tail recursion:

TAIL-RECURSIVE-QUICKSORT (*A, p, r* )

1 **while** p < r

2 **//** Partition and sort left subarray.

3 q = PARTITION(*A, p, r* )

4 TAIL-RECURSIVE-QUICKSORT(*A, p, q-1* )

5 p = q + 1

***a.*** Argue that TAIL-RECURSIVE-QUICKSORT(*A, 1, A.length* )correctly sorts the array A.

Compilers usually execute recursive procedures by using a ***stack*** that contains pertinent information, including the parameter values, for each recursive call. The information for the most recent call is at the top of the stack, and the information for the initial call is at the bottom. Upon calling a procedure, its information is ***pushed*** onto the stack; when it terminates, its information is ***popped***. Since we assume that array parameters are represented by pointers, the information for each procedure call on the stack requires stack space. The ***stack depth*** is the maximum amount of stack space used at any time during a computation.

***b.*** Describe a scenario in which TAIL-RECURSIVE-QUICKSORT’s stack depth is on an -element input array.

***c.*** Modify the code for TAIL-RECURSIVE-QUICKSORT so that the worst-case stack depth is . Maintain the expected running time of the algorithm.

6. Use the Master Theorem and the recurrence (2.1) on page 36 to prove that Merge-Sort has runtime .

7. Prove correctness (including halting) of SelectionSort (use loop invariants).

8. Provide worst and average case runtime analysis of SelectionSort.

9. Provide runspace analysis of SelectionSort.