UNIVERSITY OF CALIFORNIA
Department of Electrical Engineering
and Computer Sciences
Computer Science Division

CS61B **M. Brudno**
Summer 2000

Homework #6

Due: Wednesday, August 2nd, 2000

All of the problems are only written. Use the command '`submit hw6`' from within that directory. Put your answers to the questions in a file called `hw6.txt`.

1. The quicksort algorithm in Chapter 8 of Goodrich & Tamassia contains two recursive calls to itself. After partitioning the array, the left array is recursively sorted and then the right sub-array is recursively sorted. The second recursive call to Quicksort is not really necessary; it can be avoided by using an iterative control structure. This technique, called *tail recursion*, is provided automatically by good compilers.

a. Implement a quicksort algorithmthat makes only one recursive call to itself. Assume that `int getPivot(int[] a)` and `void partition(int[] a)` are defined.

Compilers usually execute recursive procedures by using a stack that contains pertinent information, including parameter values, for each recursive call. When a method is invoked, its information is pushed onto the stack; when it terminates, its information is popped. Since the array is represented by a reference, the information for each procedure call on the stack requires O(1) 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 the stack depth of `quicksort_t` is $\Theta(n)$ on an n-element input array.

c. Modify the code of `quicksort_t` so that the worst-case stack depth is $\Theta(\log n)$.

2. Although quicksort is a great general-purpose sort, it is no longer the best if you have additional information about the data:

a. Let's say the n points are all within the unit circle (that is the circle with radius 1, centered around the origin). Sort the n points by their distances $d_i = \sqrt{x_i^2 + y_i^2}$ from the origin in $\Theta(n)$ expected-time.

b. Dutch Flag sort: Soccer fans know that the Dutch national flag has 3 colors: Orange, White, and Blue. You are given an array where every element is one of these three values. Write a Java program (not pseudo-code) which will sort this array (Orange ¡ White ¡ Blue) In O(n) time, with a further restriction that every element of the array can only be looked at once (You are allowed to look many times in the same cell, but only if you are sure it contains a different element).

c. In some cases using the Dutch Flag sort for partitioning the elements of an array while doing quicksort will be much more efficient than using the standard partitioning scheme shown in lecture. What are those cases?

3. GT p. 339 C-8.13, C-8.16, C-8.17