

**Graduate Comprehensive Exam  
Data Structures and Algorithms  
Spring 2001**

Throughout the following you may use C, C++, Ada, or pseudocode (with sufficient details) in your answer to any question where appropriate. However, in each case be sure to specify which language you are using.

1. (25 points) Suppose that an array is used to store a set of integers (positive and negative) where, for a set of size  $n$ , the  $n$  integers are stored in array locations 1 through  $n$ .

(a) Give a function that will have two parameters: an array  $A$  containing a set of integers, and an integer  $x$ . The function will return *true* if the array contains two elements that sum to  $x$ , and *false* otherwise.

(b) What is the worst case running time of your function in part (a)?

(c) Give a function that will have two arrays as parameters, each containing a set of integers. The function will return *true* if the sets are disjoint and *false* otherwise (Recall that by definition a set contains no duplicates).

(d) What is the worst case running time of your function in part (c)? For this question you can assume that the sizes of the two sets are  $m$  and  $n$ .

2. (20 points) Euclid proved that the gcd of two numbers  $m$  and  $n$  can be computed as follows. If either  $m$  or  $n$  is equal to 0, then the gcd of  $m$  and  $n$  is the other number. For example, if  $m=0$ , then the gcd of  $m$  and  $n$  is  $n$ . On the other hand, if neither  $m$  nor  $n$  is zero, then the gcd of  $m$  and  $n$  is the same as the gcd of the smallest of these two numbers and the remainder of the division of the largest by the smallest. Give a *recursive* function that will have two positive integers  $m$  and  $n$  as parameters, and will compute and return the gcd of the two integers based on Euclid's algorithm (all non-recursive solutions will receive partial credit).

3. (15 points) What is the worst case running time of the following function?

```
void proc (int x)
{
    int i;

    if (x >= 1) {
        proc(x-1);           // first recursive call
        proc(x-1);           // second recursive call
        for (i = 0; i < 10; i++) // loop 10 times
            print(i);
    }
}
```

4) (20 points) Trees

(a) How many nodes are in a complete binary tree of height  $h$ ?

(b) How many leafs are in a complete binary tree of height  $h$ ?

(c) How many edges are in a tree with  $n$  nodes?

(c) Give a lower bound on the height of a binary tree with  $n$  leaf nodes.

5. (20 points) Graphs

(a) Describe a data structure for representing an undirected graph. For this question you can use class or type declarations for convenience, but you are not required to do so.

(b) Describe another data structure for representing an undirected graph, which is (non-trivially) different from your answer to part (a).

(c) Compare your answers to parts (a) and (b). Under what circumstances would one be preferable to the other? What are their advantages and disadvantages?