CSCI 539 Algorithms

Homework 2

Due: October 11, 2001

1. For each of the following program fragments, give an analysis of the running time. You may use summations to evaluate the running times of nested loops.

   (a)
   ```
   sum = 0
   for i = 1 to n
      for j = 1 to i * i
         for k = 1 to j
            sum ++
   ```

   (b)
   ```
   sum = 0
   for i = 1 to n
      for j = 1 to i * i
         if j mod i == 0
           for k = 1 to j
              sum ++
   ```

2. For each pair of functions (A,B) below, indicate whether A is O, Ω, or Θ of B. Note that more than one of these relations may hold for a given pair; list all correct ones. No explanation is necessary.

   (a) $(A,B) = ((\log n)^{10}, n^{0.01})$

   (b) $(A,B) = (\log(n!), \log(n^n))$

   (c) $(A,B) = (4^n, 2^n)$

   (d) $(A,B) = (n^{\frac{1}{\log n}}, 2^{\sqrt{2\log n}})$

3. Let A be an array of positive or negative integers of size n, where $A[1] < A[2] < \cdots < A[n]$. Design an $O(\log n)$ algorithm to find an i such that $A[i] = i$ provided such an i exists. Your algorithm should return 0 if such an i does not exist.

4. An array $A[1 \ldots n]$ contains all the integers from 0 to n except one. It would be easy to determine the missing integer in $O(n)$ time by using an auxiliary array $B[0 \ldots n]$ to record which numbers appear in A. In this problem, however, we cannot access an entire integer in A with a single operation. The elements of A are represented in binary, and the only operation we can use to access them is "fetch the jth bit of $A[i]$", which takes constant time. Show that if we use

only this operation, we can still determine the missing integer in $O(n)$ time. (Hint: Use divide-and-conquer. After one linear scan of the list, one can get a subproblem with size at most half of the original size.)