# CSE 373, Spring 2004, Assignment 2

April 13, 2004

Do this assignment individually. Turn in hardcopy in class on Friday, April 16.

1. Give an inductive proof that for all $n > 0$,

$$\sum_{i=1}^{n}(i+1) = \frac{n(n+3)}{2}$$

2. For the table below, we compare pairs of functions $f(n)$ (shown on the left) and $g(n)$ (shown at the top). For each cell, fill in the strongest true relationship between $f$ and $g$:

   $O$ if $f(n)$ is $O(g(n))$
   $\Omega$ if $f(n)$ is $\Omega(g(n))$
   $\Theta$ if $f(n)$ is $\Theta(g(n))$
   $-$ if none of the above.

   If there is a * in the box, then also give a short explanation of why that relationship holds.

| | 100 | $2n+5$ | $\log_2 n$ | $5n^2$ | $n \log_2 n$ |
|---|---|---|---|---|---|
| $3n+1$ | | | | | |
| $0.001 * 2^{n-10}$ | | | | * | |
| $\log_{10} n^n$ | | | | | * |

3. (a). Write an algorithm in pseudocode that uses two stack objects Sa and Sb to implement a queue ADT. You should handle the methods isEmpty, enqueue and dequeue. Assume the stack supports isEmpty, push, and pop.

   (b). What is the time complexity (worst case) for each of the queue's methods and why?

   (c). Suppose $n$ items are enqueued and dequeued in some arbitrary order. What the time complexity for this entire sequence of operations? Why?