MAT385 Test 2 (Spring 2009): Induction, Recursion, Sets, Relations

Name:

Directions: I've noticed that some of you aren't doing your homework. The following six problems are taken directly from your homeworks. I hope that this will provide motivation to pay attention more attention to it.

Problems are equally weighted. Show your work! Answers without justification will likely result in few points. Your written work also allows me the option of giving you partial credit in the event of an incorrect final answer (but good reasoning). Indicate clearly your answer to each problem (e.g., put a box around it). **Good luck!**

Problem 1. (10 pts) Use mathematical induction to prove the following about the sum of the first n triangular numbers (for every positive integer n):

$$1 + 3 + 6 + \ldots + \frac{n(n+1)}{2} = \frac{n(n+1)(n+2)}{6}$$

Problem 2. (10 pts) Prove that, for $n \geq 1$,

$$F(n+3) = 2F(n+1) + F(n)$$

where $F(n)$ is the n^{th} Fibonacci number. Do this in two different ways:

1. by the recursive definition of the Fibonacci numbers;

2. by the second principle of mathematical induction.

Problem 3. (10 pts) Solve the following recurrence relation subject to the initial conditions given:

$$
\begin{aligned}
&B(1) = 3 \\
&B(2) = 14 \\
&B(n) = 4B(n-1) - 4B(n-2), n \geq 3
\end{aligned}
$$

Problem 4. (10 pts) Bubble sort works by making repeated passes through a list; on each pass, adjacent elements that are out of order are exchanged. At the end of pass 1, the maximum element has "bubbled up" to the end of the list and does not participate in subsequent passes. The following algorithm is called initially with $j = n$.

```
BubbleSort(list L; integer j)
  // Recursively sorts the items from 1 to j in list L into increasing order
  if j=1 then
    sort is complete, write out the sorted list
  else
    for i=1 to j-1 do
      if L[i] > L[i+1] then
        exchange L[i] and L[i+1]
      end if
    end for
    BubbleSort(L,j-1)
  end if
end function BubbleSort
```

1. Walk through the bubble sort algorithm to sort the list 5, 6, 3, 4, 8, 2.

2. Write a recurrence relation for the number of comparisons of list elements done by this algorithm to sort an n-element list.

3. Solve this recurrence relation to produce a closed form solution.

Problem 5. (10 pts) Prove that if $A \subseteq B$ then $\mathcal{P}(A) \subseteq \mathcal{P}(B)$.

Problem 6. (10 pts) Consider the RNA codons that code for amino acids (composed of the bases A, C, G, and U). Describe the equivalence classes formed by the following:

1. The two codons have the same number of As (e.g., $AUA \sim GAA$).

2. The two codons have the same number of Cs or Us (e.g., $ACU \sim GUU$).