CMSC351 - Fall 2014, Homework #1

Due: Sept 22nd at the start of class

PRINT Name:

- Grades depend on neatness and clarity.
- Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should ALWAYS prove the correctness of your algorithms either directly or by referring to a proof in the book.
- Write your answers in the spaces provided. If needed, attach other pages.
- The grades would be out of 100. Four problems would be selected and everyones grade would be based only on those problems. You will also get 25 bonus points for trying to solve all problems.

Problem 1 For every integer $n \geq 1$, prove by induction that $1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$.

Problem 2 For every integer $n \geq 1$, prove by induction that $1 \times 1! + 2 \times 2! + \cdots + n \times n! = (n+1)! - 1$.

Recall that $n! = 1 \times 2 \times 3 \times \ldots n$.

Problem 3 We have $n \times k$ cups. Each of these cups has one of the k different colors. Assume k boxes of size n are available for packing these cups. Prove we can pack these cups in a way that each box has cups of at most two different colors. (Hint: Use induction on k.)

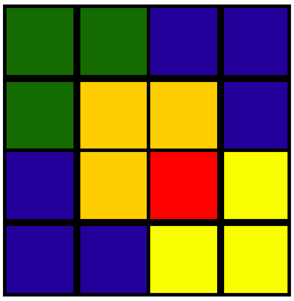

Figure 1: This board with one cell removed is fully covered by L-tromioes.

Problem 4 Suppose we have a 2^n by 2^n chessboard with one cell removed. Prove that the board can be covered with L-trominoes.

For example if $n = 2$ and one of the center cells is removed, the tiling is as shown in figure 1.

Problem 5 Consider the following code:

```
s ← 0
c ← 0
for i ← 1 to n do
    for j ← 2i to 2n do
        s ← s + j
        c ← c + 1
    end for
end for
```

(a) What is s after running this code?

(b) What is c after running this code?

(c) What is the running time of the algorithm?

Problem 6 The following code gets n and c as input. What is the running time of the following code?

```
i ← c
while i < n do
    i ← i × c.
end while
```

Problem 7 The following code represents a version of the well-known sorting algorithm "Bubble Sort":

```
for i ← 1 to n do
    for j ← 1 to n − i do
        if A[j] > A[j + 1] then
            Swap(A[j], A[j + 1])
        end if
    end for
end for
```

(a) Prove that at the end of the first iteration of the outer for loop, which is indexed by i, the largest element of the array would be at its correct position which is the end of the array.

(b) Prove by induction the correctness of the algorithm.

(c) What is the running time of Bubble Sort?

Problem 8 Order the following pairs of functions in terms of order of magnitude. In each case, briefly explain whether $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, and/or $f(n) = \Theta(g(n))$.

(a) $f(n) = n^3 + 3n + log(n)^5, \qquad g(n) = 100n^2 + 3n^3 + 10\sqrt{n}$

(b) $f(n) = \sqrt{n}\ log(n), \qquad g(n) = 2^{\sqrt{n}}$

(c) $f(n) = log(log(n^2)), \qquad g(n) = log(log(3n + 10))$

(d) $f(n) = 3^n, \qquad g(n) = n^2 2^n$

(e) $f(n) = n^{\frac{1}{2}} \qquad g(n) = n^{\frac{1}{3}} log(n)$

Problem 9 Prove the following:

(a) $n! = o(n^n)$

(b) $log(n!) = \Theta(log(n^n))$.