HW1

Due: Fri. May 22nd, 11:55 pm

This is an individual assignment. Please review the rules on academic misconduct on the course D2L page. Note that you are not required to write any code for this assignment. However, the rules on academic misconduct apply to both written and programming assignments. Here is what it boils down to:
DO NOT share your answers with others in any way (in person or electronically); DO NOT ask others to provide you with answers; write up the solutions yourself and clearly cite any external sources (books, websites, forums, discussions with peers etc.) that helped you arrive at your solutions.
If you have any doubts, please contact the instructor.

The purpose of this homework assignment is to give you additional practice with asymptotic notations and proofs of correctness. You are asked to solve a few problems that are similar to the problems discussed in lectures and live sessions. Please write the solutions to these problems clearly showing all the work you did to arrive at a solution. For problems that ask you to show or prove something, please write your proofs succinctly so that your reasoning is easy to follow. Be precise and do not skip any non-trivial steps. Please also pay attention to punctuation and grammar.

Please submit your solution via gradescope.

1. Asymptotic Notations (10 points):
   (a) Prove that $\sum_{i=1}^{n} i^k = \Theta(n^{k+1})$.
   (b) Prove that little-o is transitive, i.e. if $f(n) = o(g(n))$ and $g(n) = o(h(n))$, then $f(n) = o(h(n))$.
   (c) Let $p(n) = a_0 + a_1 n + a_2 n^2 + \cdots + a_k n^k$ where $a_k > 0$. Show that $p(n) = \Theta(n^k)$.
   (d) Find two positive valued functions $f(n)$ and $g(n)$ such that neither $f(n) = O(g(n))$ nor $g(n) = O(f(n))$.

2. (10 points)
   Using a loop invariant or otherwise, determine the asymptotic complexity of the following loop. Full points will only be awarded for the most precise asymptotic characterization. Please show all the steps that lead you to the asymptotic characterization.

```
sum = 0;
for( i = 1; i < n; i++ )
  for( j = 1; j < i * i; j++ )
    if( j % i == 0 )
      for( k = 0; k < j; k++ )
        sum++;
```

3. (10 points)
   CLRS (Problem 2-3):
   Horner's rule is an efficient method to evaluate a univariate polynomial $f(x) = \sum_{k=0}^{n} a_k x^k$. It is given by the following algorithm:

```
$y \leftarrow 0$;
for $i \leftarrow n$ to $0$ do
    $y \leftarrow a_i + x \cdot y$ ;
end
```

(a) Trace the execution of this algorithm for $x = 3$ and $f(x) = 4x^4 + 8x^3 + x + 2$.

(b) Consider the following loop invariant:
At the start of each iteration of the for loop:

$$y = \sum_{k=0}^{n-(i+1)} a_{k+i+1}x^k.$$

Show that this loop invariant holds by showing the *initialization* and *maintenance* steps.

(c) Use the loop invariant above to show the partial correctness of this algorithm.

(d) Analyze the running time of Horner's rule and state your answer using the most precise asymptotic notation.