Sections: **Worksheet Week 4 Thursday** Date:

Your TA may or may not give you specific advice or directions on which questions to try first.

Exercise 1.
For following each pair of functions $f, g : \mathbb{Z}^+ \to \mathbb{R}^+$, determine whether $f = O(g)$, $f = \Omega(g)$ or $f = \theta(g)$ is true.

(a) $f(n) = 100n^3 + 10n^2 + n + 1$, $g(n) = n^3 \log(n)$.

(b) $f(n) = n^3 - 10n - 1$, $g(n) = n^3$.

(c) $f(n) = \log_5 n$, $g(n) = \log_7 n + \log_3 n$.

Exercise 2.
Write down the asymptotic growth of the following functions.

(a) $f(n) = \log\log\log n + \log\log n + \log n$.

(b) $f(n) = n \log n + \sqrt{n^3}$.

(c) $f(n) = 10^n + n!$.

(d) $f(n) = \frac{n^3+2n}{n^2-\log n}$.

Exercise 3.

(a) Below is a baby version of an algorithm to determine whether a natural number is prime:

Input: n, a natural number
If $(n = 1)$ Return(False)
If $(n \leq 3)$ Return(True)
If $(2 \mid n)$ Return(False)
For $(i = 3, i < n, i := i + 2)$
  if $(i \mid n)$ Return(False)
End-for
Return(True)

What is the complexity?

(b) There are lots of ways to improve above code, one easy way is to observe that a prime factor of n must be less than or equal to $\sqrt{n}$, so instead of requiring $i \leq n$, we might replace it by $i \leq \sqrt{n}$. What is the complexity now?

Exercise 4.
Briefly describe what's the meaning of following algorithm, and what's the complexity.

Input: N, a natural number.

Output: c.

$c = 0$.

For$(i = 1, i < N, i++)$

  For $(j = 1, j < N - i, j++)$

    For $(k = 1, k \leq N - i - j, k++)$

      If $i^2 = j^2 + k^2$, $c := c + 1$

    End-For

  End-For

End-For

Return$(c/2)$

Exercise 5.
Design a FSM that accepts all binary strings of odd length.

Exercise 6.
Design a FSM that accepts all binary strings with no consecutive 0's nor 1's.