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 $(n \le 3)$ 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 $i \le n$, we might replace it by $i \le \sqrt{n}$. 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.

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

3min

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

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

$f(n) < n^3 \log(n)$ $n >> 0$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{100n^3 + 10n^2 + n + 1}{n^3 \log(n)}$$

$$= \lim_{n\to\infty} \frac{100 + \frac{10}{n} + \frac{1}{n^2} + \frac{1}{n^3}}{\log(n)}$$

$$= 0 \qquad f = O(g)$$

For any $c \in \mathbb{R}^{>0}$

$$\lim_{n\to\infty} \frac{f(n)}{cg(n)} = 0 \Rightarrow \frac{f(n)}{cg(n)} < 1 \text{ when } n >> 0$$

$$\Rightarrow f(n) < cg(n) \quad n >> 0$$

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

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^3 - 10n - 1}{n^3} = 1$$

$f(n) < g(n)$: $n^3 - 10n - 1 < n^3$ $\quad f = O(g)$

$\frac{1}{2} g(n) < f(n)$: $\frac{1}{2} n^3 < n^3 - 10n - 1$ $\quad f = \Omega(g)$

$\Longleftrightarrow 10n + 1 < \frac{1}{2} n^3$ $\quad n >> 0$

$f = \Theta(g)$

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

$$\log_a b = \frac{\ln b}{\ln a} \qquad f(n) = \frac{\ln(n)}{\ln(5)} \qquad g(n) = \left(\frac{1}{\ln 7} + \frac{1}{\ln 3}\right)\ln(n)$$

$$f(n) = \Theta(g(n)$$

Exercise 2.

Write down the asymptotic growth of the following functions.

5min

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

$\log x < x \qquad x > 0$

$\log\log n < \log n$

$\log\log\log n < \log\log n < \log n$

$f(n) < 3\log(n) \qquad n >> 0$

$\log n < f(n) \qquad n >> 0$

$f(n) = \Theta(\log n)$

(b) $f(n) = n\log n + n^{\frac{3}{2}} = n(\log n + n^{\frac{1}{2}})$.

$\log x < x^{\alpha} \quad \alpha > 0 \quad x >> 0$

$F(x) = \frac{\log x}{x^{\alpha}} \qquad \lim_{x\to\infty} F(x) = \lim_{x\to\infty} \frac{\log x}{x^{\alpha}}$

$= \lim_{x\to\infty} \frac{\frac{1}{x}}{\alpha x^{\alpha-1}} = \lim_{x\to\infty} \frac{1}{\alpha x^{\alpha}} = 0$

$f = \Theta(\sqrt{n^3})$.

(c) $10^n << n! \quad n >> 0$

$\underbrace{10 \times 10 \times 10 \cdots \times 10}_{n} << \underbrace{1 \times 2 \times \cdots \times 10}\times 11 \times \cdots \times n$

$$f(n) = \Theta(n!)$$

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

$$\lim_{n \to \infty} \frac{f(n)}{n} = \lim_{n \to \infty} \frac{n^3 + 2n}{n(n^2 - \log n)} = \lim_{n \to \infty} \frac{1 + \frac{2}{n^2}}{1 - \frac{\log n}{n^3}} = 1$$

$$f(n) = \Theta(n)$$

Exercise 3. 5 min

(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 $(i = 3, i < \cancel{n}\ \sqrt{n}, i := i + 2)$

  if $(i \mid n)$ Return(False)

End-for

Return(True)

+ constants

$2 \mid n$

2 divides n.

$\iff$ n has factor 2

$\frac{n}{2}$.

$\Theta(\frac{n}{2}) = \Theta(n)$.

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? $\Theta(\frac{\sqrt{n}}{2}) = \Theta(\sqrt{n})$

Exercise 4. 10min

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

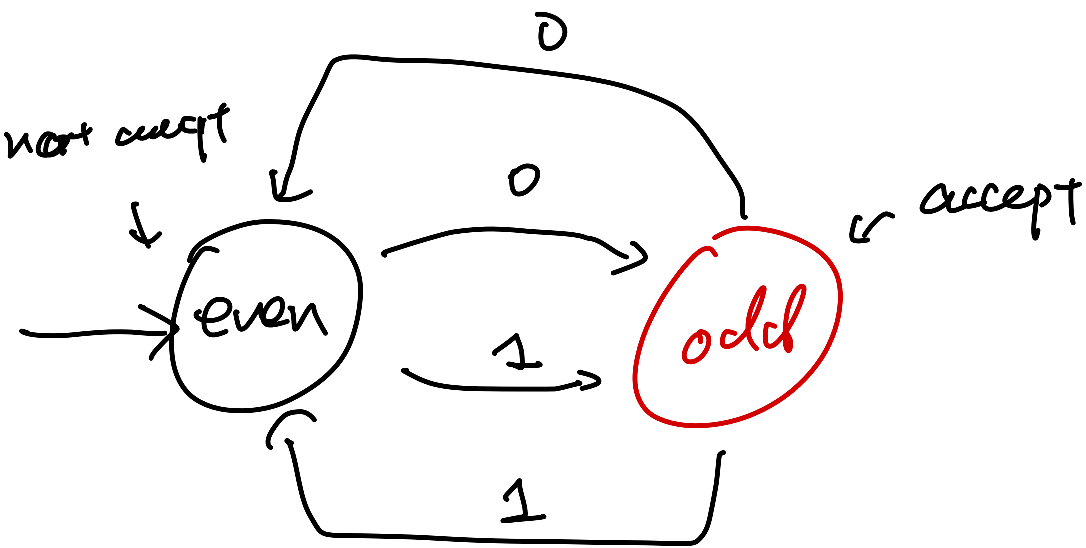
End-For

End-For

Return$(c/2)$

Exercise 5. 5min

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



$$n = p_1 \cdot p_2 \cdots \cdot p_k$$

$$p_1 \le p_2 \le \cdots \le p_k$$

If n is not prime, then $k \geq 2$

$$n \geq p_1 \cdot p_1 \cdots p_1 \geq p_1^2$$

$$p_1 \le \sqrt{n}$$

Exercise 6. 5min

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