CSCI 3104: Algorithms
Spring 2022
Recitation #5 - Asymptotics

It may be helpful to recall the following:

(Limit Comparison Test) If the limit $L := \lim_{x\to\infty} \frac{f(x)}{g(x)}$ exists, then:

- if $0 < L < \infty$, then $f(x) \in \Theta(g(x))$
- if $L = 0$, then $f(x) \in O(g(x))$, but $f(x) \notin \Theta(g(x))$
- if $L = \infty$, then $g(x) \in O(f(x))$, but $g(x) \notin \Theta(f(x))$

(L'Hôpital's rule) Suppose g and f are both differentiable functions, with either

a. $\lim_{x\to\infty} f(x) = 0$ and $\lim_{x\to\infty} g(x) = 0$; or

b. $\lim_{x\to\infty} f(x) = \pm\infty$ and $\lim_{x\to\infty} g(x) = \pm\infty$

If $\lim_{x\to\infty} \frac{f(x)}{g(x)}$ exists, then $\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$.

Problem 1

Order the following functions in increasing growth rate, using Big-O and Big-Theta notation. (e.g., $n \in O(n^2)$, $n^2 \in \Theta(n^2+1)$, $n^2+1 \in O(n^3)$)

$$n^3 + 8n^2, \quad 2n^2\sqrt{n}, \quad 6n^3 - n^2, \quad n^2 + 300n$$

Problem 2

Simplify the following expressions such that a) all logarithms are base 2 and b) terms are expressed as sums and multiplication where possible (remove exponents). It may be helpful to recall the following:

$$\log_b(mn) = \log_b(m) + \log_b(n), \qquad \log_b\left(\frac{m}{n}\right) = \log_b(m) - \log_b(n)$$
$$\log_b(n^p) = p\log_b(n), \qquad \log_b(n) = \frac{log_p(n)}{log_p(b)}$$

a. $\log_3(x^2y^3) - \log_2(\sqrt{z})$

b. $\log_3\left(\frac{(t+5)}{(t^4)}\right)$

c. $\log_2(2^x) + \log_4(\frac{x}{5y})$

Problem 3

Order the following problems in increasing growth rate, using Big-O and Big-Theta notation.

$$n \ln(2n), \quad 2^{2n}, \quad n^3 + 7, \quad n^3 + 2^{5n}$$

It may be helpful to remember the product rule: $\frac{d}{dx} f(x)g(x) = f'(x)g(x) + f(x)g'(x)$.

Problem 4

Analyze the worst-case runtime of the following algorithms. Derive the runtime complexity $T(n)$ and give an asymptotic upper bound $f(n)$ such that $T(n) = O(f(n))$.

a.
```
procedure algs_are_fun_1(integer n):
  for i = n, i >= 1
      i = i - 1
      print ‘outer’
      for j = 1, j <= 500
          j = j + 2
          print ‘inner 1’
      for k = 1, k <= 2i
          k = k + 1
           print ‘inner 2’
```

b.
```
procedure algs_are_fun_2(integer n):
  for i = 1, i <= n
      i = i + 2
      print ‘outer’
      for j = 1, j <= i
          j = 2j
          print ‘inner’
```

Problem 1

Order the following functions in increasing growth rate, using Big-O and Big-Theta notation.
(e.g., $n \in O(n^2)$, $n^2 \in \Theta(n^2+1)$, $n^2+1 \in O(n^3)$)

$$n^3 + 8n^2, \quad 2n^2\sqrt{n}, \quad 6n^3 - n^2, \quad n^2 + 300n$$

Problem 2

Simplify the following expressions such that a) all logarithms are base 2 and b) terms are expressed as sums and multiplication where possible (remove exponents). It may be helpful to recall the following:

$$\log_b(mn) = \log_b(m) + \log_b(n), \quad \log_b\left(\frac{m}{n}\right) = \log_b(m) - \log_b(n)$$
$$\log_b(n^p) = p\log_b(n), \quad \log_b(n) = \frac{log_p(n)}{log_p(b)}$$

a. $\log_3(x^2y^3) - \log_2(\sqrt{z})$

b. $\log_3\left(\frac{(t+5)}{(t^4)}\right)$

c. $\log_2(2^x) + \log_4(\frac{x}{5y})$

Problem 3

Order the following problems in increasing growth rate, using Big-O and Big-Theta notation.

$$n \ln(2n), \quad 2^{2n}, \quad n^3 + 7, \quad n^3 + 2^{5n}$$

It may be helpful to remember the product rule: $\frac{d}{dx} f(x)g(x) = f'(x)g(x) + f(x)g'(x)$.

Problem 4

Analyze the worst-case runtime of the following algorithms. Derive the runtime complexity $T(n)$ and give an asymptotic upper bound $f(n)$ such that $T(n) = O(f(n))$.

a.
```
procedure algs_are_fun_1(integer n):
  for i = n, i >= 1
      i = i - 1
      print 'outer'
      for j = 1, j <= 500
          j = j + 2
          print 'inner 1'
      for k = 1, k <= 2i
          k = k + 1
           print 'inner 2'
```

b.

```
procedure algs_are_fun_2(integer n):
  for i = 1, i <= n
      i = i + 2
      print ‘outer’
      for j = 1, j <= i
          j = 2j
          print ‘inner’
```