Advanced information, computation, communication I
EPFL - Fall semester 2020-2021

Week 7

November 3, 2020

Exercise 1.

1. Show that $5x$ is $o(x^2)$.
2. Show that $2x^2$ is not $o(x^2)$.
3. Show that $1/x$ is $o(x)$.
4. Show that if $f(x)$ is $o(g(x))$, then $f(x)$ is $O(g(x))$.

Exercise 2. Which function below grows fastest when n goes to infinity?

- ○ $(\log_3(33))^{n-3}$
- ○ 3^n
- ○ $n^{3\log_3(n)}$
- ○ $n^3 \log_3(n)$

Exercise 3. Consider the two statements below, where c and k are constants with $k \geq 2$:

n^k is $O(k^n)$ $\qquad$ $(\log n)^k e^{(c+o(1))(\log n)^{1/3}(\log(\log(n)))^{2/3}}$ is $e^{(c+o(1))(\log n)^{1/3}(\log(\log(n)))^{2/3}}$

- ○ They are both False.
- ○ Only the first is True.
- ○ Only the second is True.
- ○ They are both True.

Exercise 4. Let f be arbitrary functions from $\mathbf{N}$ to $\mathbf{R}_{>0}$.

Let g_1, g_2 be two functions from $\mathbf{N}$ to $\mathbf{R}_{>0}$ such that g_1 and g_2 are both $\Theta(f)$.

1. Show that the function $g_1 + g_2$ is $\Theta(f)$ or provide a counterexample.
2. Show that the function $g_1 g_2$ is $\Theta(f^2)$ or provide a counterexample.

Let g_3, g_4 be two functions from $\mathbf{N}$ to $\mathbf{R}$ such that g_3 and g_4 are both $\Theta(f)$.

3. Show that the function $g_3 + g_4$ is $\Theta(f)$ or provide a counterexample.
4. Show that the function $g_3 g_4$ is $\Theta(f^2)$ or provide a counterexample.

Let g be a function from $\mathbf{N}$ to $\mathbf{R}_{>0}$ such that g is $O(f)$.

5. Show that 2^g is $O(2^f)$, or provide a counterexample.

Exercise 5. Consider the two statements below, where k and ℓ are constants with $k > \ell \geq 2$ and $m \to \infty$:

$$\log_m(k) \text{ is } \Theta(\log_m(\ell)) \qquad k^{\log_\ell(m)} \text{ is } O(\ell^{\log_k(m)}).$$

- ◯ They are both false.
- ◯ Only the first is true.
- ◯ Only the second is true.
- ◯ They are both true.

Little-o. Let f and g be two functions from $\mathbf{R}$ to $\mathbf{R}$. We say that "f is little-oh of g" and write "$f(x)$ is $o(g(x))$" if:

$$\forall C \; \exists k \text{ such that } \forall x > k \;\; |f(x)| < C|g(x)|$$

where the domain for C, k, and x is $\mathbf{R}_{>0}$ (correcting the $\mathbf{R}$ that was there before). Said differently, for every constant C there exist a constant k such that $|f(x)| < C|g(x)|$ for all $x > k$. If $g(x)$ is nonzero, the little-o relation is equivalent to

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} = 0.$$

Exercise 6. Consider the following two statements:

$$\Big(f \text{ is } o(f)\Big) \quad \text{and} \quad \Big(f \text{ is } o(g) \text{ implies } f \text{ is } O(g)\Big).$$

- ◯ Only the second is true.
- ◯ They are both false.
- ◯ Only the first is true.
- ◯ They are both true.

Exercise 7. Given the two statements below, where $d > 0$ is an integer constant and a_i for all $i \in \mathbf{Z}$ are positive integers with $\max_{i \in \mathbf{Z}}(a_i) = D$ for a constant $D > 0$,

$$\sum_{i=0}^{n} a_i i^d \text{ is } \Theta(n^{d+1}) \qquad\qquad \sum_{i=0}^{d} a_i n^i \text{ is } \Theta(n^d)$$

- ◯ They are both true.
- ◯ Only the first is true.
- ◯ Only the second is true.
- ◯ They are both false.

Exercise 8. Construct two functions f and g from $\mathbf{N}$ to $\mathbf{R}_{>0}$ such that f is not $O(g)$ and g is not $O(f)$ or prove that such functions are impossible to find.

Exercise 9. Consider the following algorithm, which takes as input a sequence of n integers $a_1, a_2, ..., a_n$ and produces as output a matrix $M = \{m_{ij}\}$ where m_{ij} is the minimum term in the sequence of integers $a_i, a_{i+1}, ..., a_j$ for $j \geq i$ and $m_{ij} = 0$ otherwise.

initializ **M** so that $m_{ij} = a_i$ if $j \geq i$ and $m_{ij} = 0$ otherwise
for $i := 1$ to n
  for $j := i + 1$ to n
    for $k := i + 1$ to j
      $m_{ij} := \min(m_{ij}, a_k)$
return **M** $= \{m_{ij}\}$ $\{m_{ij}$ is the minimum term of $a_i, a_{i+1}, ..., a_j\}$

1. Show that this algorithm uses $O(n^3)$ comparisons to compute the matrix M.
2. Show that this algorithm uses $\Omega(n^3)$ comparisons to compute the matrix M. Using this fact and part (a), conclude that the algorithms uses $\Theta(n^3)$ comparisons.

Exercise 10. What is the largest n for which one can solve within a minute using an algorithm that requires $f(n)$ bit operations, where each bit operation is carried out in 10^{-12} seconds, with these functions $f(n)$?

a. *log n*

b. $1,000,000n$

c. n^2