CPSC 320: Intermediate Algorithm Design 2016 Summer Term 1

Tutorial 2

Prof. Nick Harvey *University of British Columbia*

1. **(Algebraic properties of Big-O)**
   - **(Pointwise Maximum)** Let $h = \max\{f, g\}$, i.e., $h(x) = \max f(x), g(x)$ for all x. Prove that $h = \Theta(f + g)$.
   - **(Transitivity)** Prove that: if $h = O(g)$ and $g = O(f)$, then $h = O(f)$. (Same for Ω and Θ.)
   - **(Sums)** Let k be a fixed constant. Let $f_1, ..., f_k$ and h be functions such that $f_i = O(h)$ for all i. Prove that $f_1 + \cdots + f_k = O(h)$. (Same for Ω and Θ.)
2. **(Familiar friends)**
   - **(Polynomials)** Let $T(n) = a_0 + a_1 n + \cdots + a_d n^d$, with $a_d > 0$. Then $T(n) = \Theta(n^d)$.
   - **(Base of log is irrelevant)** Prove that $\log_a n = \Omega(\log_b n)$ for all constants $a, b > 1$.
3. Rank the following functions by order of growth. That is, order them so that

   $$f_1 \in O(f_2), \quad f_2 \in O(f_3), \quad \text{etc.}$$

   Make sure to indicate whether or not $f_i \in \Theta(f_{i+1})$. For instance you could use the notation

   $$n < n^2 = 2n^2 < n^4 ...$$

   Here are the functions to be ordered:

$n^3 - n$	$n!$	$\log(\log n)$	1.01^n
$\sqrt{n \log n}$	17	$4913n^{289}$	3^{2n-17}
$1.01^{1.01^n}$	$\sqrt{n}$	3^n	n^n
$n^{18/17}$	$n \log^{17} n$	3^{2n}	$n^3 + \log n$

4. (This example uses an **important trick!**) Prove that the runtime of the following algorithm is $\Theta(n \log n)$.

   ```
   for i = 1, ..., n
       for j = 1, ..., ⌈log i⌉
           Print "Hi";
   ```

5. Let k be a fixed constant (independent of n). Prove that

   $$1^k + 2^k + \cdots + (n-1)^k + n^k \;\in\; \Theta(n^{k+1})$$

   Hint: do not use induction. Instead, do O and Ω separately.

   Hint: To handle Ω, you will need to use the same trick as the previous question.