CMPS 6610 Algorithms – Fall 18

9/3/18

1. Homework

Due **9/10/18** at the beginning of class

1. **Big-Oh implications (9 points)**
   Let f, g be two functions such that $f(n) \in O(g(n))$. In addition assume that f and g are well-behaved in the sense that $f(n) \geq 1$ and $\log g(n) \geq 1$ for n large enough. For each of the statements below, either prove that it is true or disprove it by giving a counterexample. *(Hint: For the proofs, use the definition of big-Oh.)*

   (a) $\log f(n) \in O(\log g(n))$
   (b) $2^{f(n)} \in O(2^{g(n)})$
   (c) $f(n)^2 \in O(g(n)^2)$

2. **Big-Oh ranking (12 points)**
   Rank the following twelve functions by order of growth, i.e., find an arrangement $f_1, f_2, ...$ of the functions satisfying $f_1 \in O(f_2)$, $f_2 \in O(f_3), ...$. Partition your list into equivalence classes such that f and g are in the same class if and only if $f \in \Theta(g)$. For every two functions f_i, f_j that are adjacent in your ordering, prove shortly why $f_i \in O(f_j)$. And if f and g are in the same class, prove that $f \in \Theta(g)$.

   $$\log\log n \;,\; 2^{n+3} \;,\; \log n! \;,\; \log n \;,\; n^3 \;,\; 8^n \;,\; 2^n \;,$$
   $$\log\sqrt{n} \;,\; \sqrt{\log n} \;,\; \sqrt{n} \;,\; 8^{\log n} \;,\; \log^2 n \;,$$

   A few important rules:

   - The rule of l'Hôpital:
     $$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{f'(n)}{g'(n)}$$
     if the limits exist; where f and g' are the derivatives of f and g, respectively.
   - Chain and product rules for differentiation.
   - $\log n = \log_2 n$
   - Log-rules (see chapter 3.2 in the book): $b^{\log_b a} = a$, $\log_b b = 1$, $a^{\log_b c} = c^{\log_b a}$
   - $n! \in O(n^n)$

3. **Lower bound (6 points)**
   Consider the following two problems:

   - Problem 1: Given an **unsorted** array A of n numbers, and a number x. Compute which elements in A are less than x.
   - Problem 2: Given a **sorted** array of n numbers, and a number x. Compute which elements in A are less than x.

   Provide lower bounds for comparison-based algorithms that solve these problems. Use decision trees.
   Hint: Make sure to specify how the output is represented.