CSE 6331 HOMEWORK 1

BRENDAN WHITAKER

1. *Order the given functions by asymptotic dominance. That is, produce an order $f_1(n), f_2(n), ...$ such that $f_i(n) = O(f_{i+1}(n)$.*

$$\begin{aligned} &1/n^2, 2^{23}, \log_2 \log_2(n^5 + n^2), 27\log_7(n) + \sqrt{\log_2(n)}, (log_2 n)^3, \\ &3n^{2.1} + 19n^{1.5} + \sqrt{n}, 5^{log_2 n} + \sqrt{n}, 4^{\sqrt{n}}, n!n^2, n^{2n}, 2^{2^n}. \end{aligned} \tag{1}$$

$$c, b, f, h, i, g, j, d, a, e, k \tag{2}$$

2. *Let $f(n)$ be a function defined for $\mathbb{N}$. Prove or disprove: If $f(n) \in \Theta(n^2)$, then $f(n)$ is asymptotically, monotonically non-decreasing, i.e. $f(n) \leq f(n+1)$ for all sufficiently large n.*

It is false, we can construct a piecewise function by for odd n taking $f(n) = n^2$ and taking $f(n) = f(n-1)$ for even n. Multiplying this by scalars gives us upper and lower bounds for n^2. We need to use induction to be rigorous.

Consider the function $f : \mathbb{N} \to \mathbb{N}$ given by:

$$f(n) = \begin{cases} n^2 & n \text{ is odd,} \\ f(n-1) - 1 & n \text{ is even.} \end{cases}$$

Note that this function is not asymptotically monotonically non-decreasing since for each odd integer n, $f(n+1) = f(n) - 1 < f(n)$. We prove that $f \in \Theta(n^2)$. Define $g(n) = n^2$.

We first prove that $f(n) \in O(n^2)$. Note that $\frac{1}{2}f(4) = 4$, and $g(4) = 16$. Define $h(n) = \frac{1}{2}f(n)$. We claim that $h(n) < g(n), \forall n \geq 4$. We prove this by induction. We have already checked our base case and verified that $h(4) < g(4)$. We fix $n \in \mathbb{N}$, and define our induction hypothesis as $h(n) < g(n)$. We wish to show that $h(n+1) < g(n+1)$.

Case 1: n is odd. Observe:

$$\begin{aligned} h(n+1) &= \frac{1}{2}f(n+1) = \frac{1}{2}(f(n) - 1) = \frac{1}{2}f(n) - \frac{1}{2} = h(n) - \frac{1}{2} \\ &< g(n) - \frac{1}{2} < g(n+1). \end{aligned} \tag{3}$$

In the above expression, the first equality comes from the definition of h, the second comes from the definition of f and the fact that n is odd, and the third comes from the definition of h again. The inequality on the second line comes from the induction hypothesis, and since $g(n) = n^2$ is an increasing function for positive n, we have the final inequality and our desired result for the first case.

Case 2: n is even. Observe:

$$h(n+1) = \frac{1}{2}f(n+1) = \frac{1}{2}(n+1)^2 = \frac{1}{2}g(n+1) < g(n+1). \tag{4}$$

In the above expression, the first equality comes from the definition of h, the second comes from the definition of f and the fact that n is even, and thus $n+1$ is odd. The third equality comes from the definition of g and the last is because g is positive for $n \geq 4$.

Hence we have the desired inequality for all $n \geq 4$, and the induction is complete. So we know $f(n) < 2n^2\ \forall n \geq 4$, hence $f(n) \in O(n^2)$.

Now we will show $f(n) \in \Omega(n^2)$. So we choose $c = \frac{1}{2}$, and we will show that $f(n) \geq \frac{1}{2}n^2$ for all sufficiently large n. Note that $f(5) = 25$. Define $g'(n) = \frac{1}{2}n^2$. Note that $g'(5) = 25/2 < 25 = f(5)$,

Date: SP18.

so $f(5) > g'(5)$. We prove by induction that $f(n) > g'(n)\ \forall n \geq 5$. We already finished our base case, so we fix n, and assume for our induction hypothesis that $f(n) > g'(n)$. We show $f(n+1) > g'(n+1)$.

Case 1: n is odd. Observe:

$$f(n+1) = f(n) - 1 = n^2 - 1 = (n+1)(n-1) > (n+1)(n+1) > \frac{1}{2}(n+1)^2 = g'(n+1).$$

Case 2: n is even. Observe:

$$f(n+1) = (n+1)^2 < \frac{1}{2}(n+1)^2 = g'(n+1).$$

So in both cases our inductive step holds, so we have proven that $f(n) > \frac{1}{2}n^2$, $\forall n \geq 5$. So $f(n) \in Omega(n^2)$, hence we know $f(n) \in \Theta(n^2)$. But since f is not asymptotically monotonically non-decreasing, we have found a valid counterexample.

3. *Let $f(n)$ be defined on $\mathbb{N}$. Prove or disprove the following statement: if $f(n) \in O(g(n))$, then $2^{f(n)} \in O(2^{g(n)})$.*

   Let $f(n) = 2n$, and let $g(n) = n$. Then $f(n) \in O(g(n))$, since $f(n) = 2n \leq 3g(n) = 3n\ \forall n$. But note that $2^{2n} \notin O(2^n)$, since $2^{2n} = (2^n)^2 = 4^n$, and we know $4^n \notin O(2^n)$, since $2 < 4$.

4. *Find how many dollar signs the given procedure will print.*

   Note that the inner while loop executes $\log_2(l)$ times where $2^l = n$. So we have that the inner while loop takes $\log\log(n)$ time. The outer for loop executes $\log_2(n)$ times. Hence our running time is given by:

$$T(n) = \Theta(\log(n)\log(\log(n))).$$