Anand Mahadevan - SID 862132182

CS141 Homework 1

due Tuesday, August 10th 11:59 PM

Problem 1.
Use L'Hopital's theorem to prove that:

$$log(n)^{k_1} = o(n^{k_2})$$

For any values of k1 and k2 including the case where k1 is not integer.

Answer:
$log(n)^{k_1} = o(n^{k_2})$ if $\lim_{n\to\infty} log(n)^{k_1}/n^{k_2}$ is equal to 0.

First case: k_1 **is an integer.**

$$\lim_{n\to\infty} \frac{log(n)^{k_1}}{n^{k_2}} =$$
$$\lim_{n\to\infty} \frac{k_1 * log(n)^{k_1-1} * \frac{1}{n} * \frac{1}{ln(10)}}{k_2 * n^{k_2-1}} =$$

After taking the derivative of numerator and denominator $k_1 - 1$ more times...

$$\frac{k_1!}{(k_2 * (k_2-1) * ... * (k_2-k_1+1)) * (ln(10))^{k_1}} \lim_{n\to\infty} \frac{1/n^{k_1}}{n^{k_2-k_1}} =$$
$$\lim_{n\to\infty} \frac{1}{n^{k_2-k_1} * n^{k_1}} =$$
$$\lim_{n\to\infty} \frac{1}{n^{k_2}} =$$
$$= 0$$

Q.E.D

Second case: k_1 **is not an integer.**

$$\lim_{n\to\infty} \frac{log(n)^{k_1}}{n^{k_2}} =$$
$$\lim_{n\to\infty} \frac{k_1 * log(n)^{k_1-1} * \frac{1}{n} * \frac{1}{ln(10)}}{k_2 * n^{k_2-1}} =$$

After taking the derivative of numerator and denominator $\lceil k_1 - 1 \rceil$ more times...

$$\frac{k_1 * (k_1 - 1) * ... * (k_1 - \lceil k_1 \rceil)}{k_2 * (k_2 - 1) * ... * (k_2 - \lceil k_1 \rceil + 1) * (ln(10))^{\lceil k_1 \rceil}} \lim_{n \to \infty} \frac{1/n^{\lceil k_1 \rceil}}{log(n)^{\lceil k_1 \rceil - k_1} * n^{k_2 - \lceil k_1 \rceil}} =$$

$$\lim_{n \to \infty} \frac{1}{log(n)^{\lceil k_1 \rceil - k_1} * n^{k_2 - \lceil k_1 \rceil} * n^{\lceil k_1 \rceil}} =$$

$$\lim_{n \to \infty} \frac{1}{log(n)^{\lceil k_1 \rceil - k_1} * n^{k_2}} =$$

$$= 0$$

Q.E.D

Problem 2.
Top of the list represents largest growth, with $g_1 = \Omega(g2)$. Multiple functions on the same line represent equal growth such that $f(n) = \Theta(g(n))$, and are separated by a large space between them. All functions are contained within curly brackets for clarity.

List starts here:
{ 2^{2^n} $\quad$ 2^{2^n+1} }
($\uparrow$ Constant to the power of exponential function; second function is simply 1st function times 2, thus they are both Θ of each other)
{(n+1)!} (Factorial; Same as below but with extra (n+1) term, will have larger growth $\to \infty$)
{n!} (Factorial)
{e^n} (Next 3 are exponential functions with decreasing base to the power of n)
{2^n}
{$(3/2)^n$}
{n^3} (Polynomial, power 3)
{n^2} (Polynomial, power 2)
{ $n * lg\ n$ $\quad$ $lg(n!)$ } ($lg(n!)$ is $\Theta(n\ lg\ n)$)
{n} (Polynomial, power 1)
{$(\sqrt{2})^{lg\ n}$} (Simplifies to $\Theta(\sqrt{(n)})$)
{$lg^2 n$} (Next 3 are decreasing powers of log n)
{$ln\ n$}
{$\sqrt{lg\ n}$}
{$ln\ (ln\ n)$} (log of log)
{1}($Constant$)

Problem 3.

a)

```
int max, secMax = NULL; // secMax is second max
int FIND-SECOND-LARGEST(Arr, x, r) // x is left index and r is right index
    if Arr length < 2 // must have minimum 2 elements to return 2nd largest
        Print error message; return NULL;
    if max equals NULL // first run through
        max = secMax = Arr[0]; // arbitrary initialization of max and secMax
    if x equals r and Arr[x] > max // if we have 1 element to deal with and it's > max
        secMax = max; max = Arr[x]; //send old max value to secMax
    if x < r // more than 1 element to deal with
        m = ⌊(x + r)/2⌋ // middle index
        temp1 = FIND-SECOND-LARGEST(Arr, x, m) // first half
        temp2 = FIND-SECOND-LARGEST(Arr, m+1, r) // second half

    return secMax;
```

b)

T(n) = T(n/2) + T(n/2) + c (with c being a constant > 0)

= 2 * T(n/2) + c

Using master theorem,

a = 2, b = 2, d = 0

This falls under the case, $d < log_b a$ with $0 < log_2 2 = 1$ where

T(n) = O($n^{log_b a}$) = O(n^1) = O(n)

Q.E.D.