Computer Science 308-250B
EXTRA problems on Big-*O* notations

Show the following

1. f_n **is** $O(2^n)$ where f_n is the nth Fibonacci number.

By mathematical induction. $f_n < 2^n$

Basis: $f_1 = 1 < 2 = 2^1$, $f_2 = 1 < 4 = 2^2$

Induction step:

Let n>2. Assume $f_{n-2} < 2^{n-2}$ and $f_{n-1} < 2^{n-1}$

$f_n = f_{n-1} + f_{n-2} < 2^{n-1} + 2^{n-2} = 3*2^{n-2} < 4*2^{n-2} = 2^n$.

2. n^k **is** $O(n^{\log \log n})$ for any k>0

□

$n^k < n^{\log \log n}$ iff $k < \log \log n$ iff $n > 2^{2^k}$

By setting $n_0 = 2^{2^k}$, we conclude n^k **is** $O(n^{\log \log n})$ for any k>0

□

3. a^n **is not** $O(n^k)$ for any $k>0$, $a>1$

□

By the limit rule.

$\mathrm{Lim}_{n\to\infty} \quad n^k/a^n$

$= \mathrm{Lim}_{n\to\infty} \quad kn^{k-1}/(\ln a)a^n$

$= \mathrm{Lim}_{n\to\infty} \quad k(k-1)n^{k-2}/(\ln a)^2a^n$

…

$= \mathrm{Lim}_{n\to\infty} \quad k(k-1)(k-2)\ldots 1/(\ln a)^k a^n$

$= \mathrm{Lim}_{n\to\infty} \quad k!/(\ln a)^k a^n = 0$

which implies

n^k **is** $O(a^n)$ for any $k>0$, $a>1$

a^n **is not** $O(n^k)$ for any $k>0$, $a>1$

□

4. $\log n!$ **is** $\Theta(n \log n)$ ($\log n!$ **is** $\Omega(n \log n)$ & **is** $O(n \log n)$)

□

Notice that $\log n! = \Sigma_{i=1..n} \log i < \Sigma_{i=1..n} \log n = n \log n$.

Also $\Sigma_{i=1..n} \log i > \Sigma_{i=n/2..n} \log i > \Sigma_{i=n/2..n} \log n/2$

$= n/2 \log n/2$.

The results follow.

□

5. whenever $m>k$, n^k **is** $O(n^{(m-\delta)})$, for some small $\delta>0$

□

Set $\delta = (m-k)/2 > 0$. Note that $m-\delta = (m+k)/2 > k$, since $m>k$.

Thus n^k **is** $O(n^{(m-\delta)})$.

□

6. whenever $m<k$, n^k **is** $\Omega(n^{(m+\delta)})$, for some small $\delta>0$

□

Set $\delta = (k-m)/2 > 0$. Note that $m+\delta = (k+m)/2 < k$, since $m<k$.

Thus n^k **is** $\Omega(n^{(m+\delta)})$.

□

Solve the following recurrences and express your solution with the big-Θ notation:

$$T_A(n) = \begin{cases} a & \text{if } n=1 \\ T_A(n/2) + bn & \text{if } n>1 \end{cases}$$

$$T_B(n) = \begin{cases} a & \text{if } n=1 \\ T_B(n/2) + bn^2 & \text{if } n>1 \end{cases}$$

$$T_C(n) = \begin{cases} a & \text{if } n=1 \\ 9\,T_C(n/3) + n^2 \log n + n + 2 & \text{if } n>1 \end{cases}$$

<u>Solutions</u>: We consider the general case

$$T(n) = \begin{cases} c & \text{if } n=1 \\ \alpha T(n/\beta) + f(n) & \text{if } n>1 \end{cases}$$

by comparing the special function $n^{\log_\beta \alpha}$ to f(n):

$$T_A(n) = \begin{cases} a & \text{if } n=1 \\ T_A(n/2) + bn & \text{if } n>1 \end{cases}$$

α=1, β=2, f(n)=bn, k=1

$n^{\log_\beta \alpha}$ = 1 **is** O(bn) since $\log_\beta \alpha$<k=1

and thus by the Master Method (3.)

$T_A(n)$ **is** $\Theta(f(n)) = \Theta(n)$

$$T_B(n) = \begin{cases} a & \text{if } n=1 \\ T_B(n/2) + bn^2 & \text{if } n>1 \end{cases}$$

$\alpha=1$, $\beta=2$, $f(n)=bn^2$, $k=2$

$n^{\log_\beta \alpha} = 1$ **is** $O(bn^2)$ since $\log_\beta \alpha < k=2$

and thus by the Master Method (3.)

$T_B(n)$ **is** $\Theta(f(n)) = \Theta(n^2)$

$$T_C(n) = \begin{cases} a & \text{if } n=1 \\ 9\,T_C(n/3) + n^2 \log n + n + 2 & \text{if } n>1 \end{cases}$$

$\alpha=9$, $\beta=3$, $f(n)=n^2 \log n+n+2$, $k=2$, $m=1$

$n^{\log_\beta \alpha} = n^2$ and thus

$n^{\log_\beta \alpha} \log n = n^2 \log n$ **is** $\Theta(n^2 \log n+n+2)$ $(k=2,m=1)$

and thus by the Master Method (2.)

$T_C(n)$ **is** $\Theta(n^{\log_\beta \alpha} \log^2 n) = \Theta(n^2 \log^2 n)$ $(k=2,m+1=2)$