Problem Set 1

All parts are due February 18th, 2016 at 11:59PM.

Name: Alex List

Collaborators: /

Part A

Problem 1-1.

(a) $f2 = log(4n^{n^4}) = n^4 log(n)$
$f2 > f1 \rightarrow f1 = O(f2)$
$f3 = 4log(4n)log(n) \simeq log(n)log(n) = log^2(n)$
$\lim n \rightarrow \inf f3/f2 = \lim n \rightarrow \inf log^2(n)/n^4 log(n) = \lim n \rightarrow \inf log(n)/n^4 = 0$
$\rightarrow f3 = O(f2)$
$\lim n \rightarrow \inf f3/f4 = \lim n \rightarrow \inf log^2(n)/log^4(n) = 0 \rightarrow f3 = O(f4)$
$\lim n \rightarrow \inf f5/f4 = \lim n \rightarrow \inf log^4(log(n))/log^4(n) = 0 \rightarrow f5 = O(f4)$
$\rightarrow f1 < f2 > f3 < f4 > f5$
Comparing $f1 \& f3$
$\lim n \rightarrow \inf f3/f1 = \lim n \rightarrow \inf log^2(n)/n^4 = 0 \rightarrow f3 = O(f1)$
$\rightarrow f3 < f1 < f2 \& f5 < f4 > f3$
Comparing $f2 \& f5$
$\lim n \rightarrow \inf f5/f2 = \lim n \rightarrow \inf log^4(log(n))/n^4 log(n) = \lim n \rightarrow \inf log(log(n))/nlog(n) = 0 \rightarrow f5 = O(f2)$
$\rightarrow f3 < f1 < f2 \& f2 > f5 < f4 > f3$
Comparing $f3 \& f5$
$\lim n \rightarrow \inf f5/f3 = \lim n \rightarrow \inf log^4(log(n))/log^2(n) = \lim n \rightarrow \inf log(log(n))/log(n) = 0 \rightarrow f5 = O(f3)$
$\rightarrow f5 < f3 < f1 < f2 \& f4 > f3 \& f4 > f5$
Comparing $f4 \& f1$
$\lim n \rightarrow \inf f4/f1 = \lim n \rightarrow \inf log^4(n)/n^4 = 0 \rightarrow f4 = O(f1)$
$\rightarrow f5 < f3 < f4 < f1 < f2$
$f5, f3, f4, f1, f2$

(b) Transform each function with logarithm and compare:
$f'1 = log(4^{4^n}) = 4^n log(4)$
$f'2 = log(4^{4^{n+1}}) = 4^{n+1} log(4)$
$f'3 = log(5^{4^n}) = 4^n log(5)$
$f'4 = log(5^{4n}) = 4n log(5)$
$f'5 = log(5^{5n}) = 5n log(5)$
By inspection:
$f4 = O(f5), f1 = O(f2), f4 = O(f3), f1 = O(f3), f3 = O(f2), f5 = O(f3), f5 = O(f1), f4 = O(f1)$
$f4 < f5, f1 < f2, f1 < f3, f3 < f2, f4 < f3, f5 < f3, f5 < f1, f4 < f1$
$\rightarrow f4 < f5 < f1 < f3 < f2$
$f4, f5, f1, f3, f2$

(c) Simply first then compare
$f1 = \frac{n!}{4!(n-4)!} = O(n^4)$
$*f2 = \frac{n!}{(n/4)!(n-n/4)!} = \frac{n!}{(n/4)!(3n/4)!} = O(1)$ n ns in numerator, n ns in denominator.
$f3 = 4n! = O(n^n)$
$f4 = 4^{n/4}$
$f5 = (n/4)^{n/4}$
Visually:
$f2 < f1 < f3, f5 = O(f3), f1 = O(f5), f2 = O(f4)$
$\rightarrow f2 < f1 < f5 < f3, f2 = O(f4)$
I'll compare f4 and f3. I wasn't sure, so I use L'Hospital's rule
$\lim n \rightarrow \inf \frac{d}{dn}\frac{f4}{f3} = \lim n \rightarrow \inf \frac{d}{dn}\frac{4^{n/4}}{n^n} = \lim n \rightarrow \inf \frac{2^{n/2-1} log(2)}{n^n (log(n)+1)} = 0$
$\rightarrow f4 = O(f3)$
I still need to compare f4 and f5. I'll use L'Hospital's rule again.
$\lim n \rightarrow \inf \frac{d}{dn}\frac{f5}{f4} = \lim n \rightarrow \inf \frac{d}{dn}\frac{(n/4)^{n/4}}{4^{n/4}} = \lim n \rightarrow \inf \frac{2^{-n/2-2} n^{n/4} log(n/4+1)}{2^{n/2-1} log(2)} = 0$
$\rightarrow f5 = O(f4)$
Because the $2^{-n/2-2}$ term goes to 0.
$\rightarrow f2 < f1 < f5 < f4 < f3$
$f2, f1, f5, f4, f3$

Problem 1-2.

(a)
1. $\theta(n)$ Branch factor 1. N iterations of c work.
2. $O(n^2)$ Because summation over work on all levels from c to nc is $\frac{n*(n+1)}{2} = \frac{n^2+n}{2} = O(n^2)$

3. $\theta(log(n))$ Because $log(n)$ levels with c work.
4. * $O(n)$ Because with branching factor 2, work c per node, bottom row does most work, $c * 2^{height=log(n)} = cn = O(n)$
5. $\theta(nlog(n))$ Because there are $log(n)$ rows of cn work per row.
6. * $O(n^{log(3)})$ Because with branching factor 3, the bottom row does most work $3^{height=log(n)}$ nodes of c work $= c * n^{log(3)/log(2)} = O(n^{log(3)})$

(b)
1. $T(n) = T(n/2) + c$ Binary search has one subproblem, with half the nodes to visit. $O(log(n))$ tota.
2. * What was name of the matrix algorithm? $T(n) = T(n/2) + c * log(n)$ For nxn matrix, Instead of doing c work for 1 comparison, I search the n-width row, $log(n)$ work per subproblem.

Problem 1-3.

(a)

(b)

Part B

Problem 1-4.

Submit your implemented python script.

(a)

(b)

(c)

(d)

(e)