Homework 2 (90 points), Spring 2004

Q1: Exercise 2.3-3 (10 points)

Use mathematical induction to show that when n is an exact power of 2, the solution of the recurrence

$$T(n) = \begin{cases} 2 & \text{if } n = 2 \\ 2T(n/2) + n & \text{if } n = 2^k, \text{ for } k > 1 \end{cases}$$

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

- Base Step:
  If $n = 2$, then $T(2) = 2$ and $2 \log 2 = 2$
  Thus, $T(2) = 2 \log 2$

- Hypothesis Step:
  Assuming $T(n) = n \log n$ is true if $n = 2^k$ for some integer $k > 0$

- Induction Step:
  If $n = 2^{k+1}$, then

$$\begin{aligned} & T(2^{k+1}) \\ = & \ 2T(2^{k+1}/2) + 2^{k+1} \\ = & \ 2T(2^k) + 2^{k+1} \\ = & \ 2(2^k \log 2^k) + 2^{k+1} \\ = & \ 2^{k+1}((\log 2^k) + 1) \\ = & \ 2^{k+1} \log 2^{k+1} \end{aligned}$$

Q2: Exercise 3.1-3 (10 points)

Explain why the statement, "The running time of algorithm A is at least $O(n^2)$," is meaningless.

- Let $T(n)$ be the running time for algorithm A and let a function $f(n) = O(n^2)$. The statement says that $T(n)$ is at least $O(n^2)$. That is, $T(n)$ is an upper bound of $f(n)$. Since $f(n)$ could be any function "smaller" than n^2 (including constant function), we can rephase the statement as "The running time of algorithm A is at least constant." This is meaningless because the running time for every algorithm is at least constant.

Q3: Problem 3.2 (10 points)

Indicate, for each pair of expressions (A, B) in the table below, whether A is O, o, Ω, ω, or Θ of B. Assume that $k \geq 1, \epsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of table with "yes" or "no" written in each box.

-

A	B	O	o	Ω	ω	Θ
$\lg^k n$	n^ϵ	yes	yes	no	no	no
n^k	c^n	yes	yes	no	no	no
$\sqrt{n}$	$n^{\sin n}$	no	no	no	no	no
2^n	$2^{n/2}$	no	no	yes	yes	no
$n^{\lg m}$	$m^{\lg n}$	yes	no	yes	no	yes
$\lg(n!)$	$\lg(n^n)$	yes	no	yes	no	yes

Q4: Exercise 4.1-6 (10 points)

Solve the recurrence $T(n) = 2T(\sqrt{n}) + 1$ by changing the variable. Your solution should be asymptotically tight. Do not worry about whether values are integral.

- Let $m = \lg n$
  Then $T(n) = T(2^m) = 2T(2^{m/2}) + 1$
  Let $S(m) = T(2^m)$
  Then $S(m) = 2T(2^{m/2}) + 1 = 2S(m/2) + 1$

  By using master method's first case, we $S(m) = \Theta(m)$
  So $T(n) = T(2^m) = S(m) = \Theta(m) = \Theta(\lg n)$

Q5: (10 points)

Use the substitution method to show the solution of $T(n) = T(n/3) + 1$ is $O(\log n)$. You may want to prove $T(n) \leq c \cdot \log_3 n$ and use $n = 3$ to show the boundary (base) case.

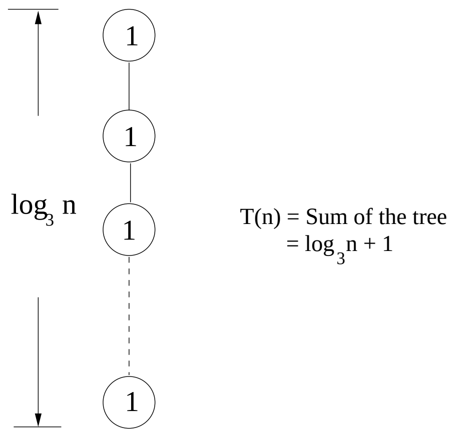
- Boundary (Base) case: When $n = 3$, $T(3) = 2 \leq c \cdot \log_3 3$ is trure if $c \geq 2$.
  Assume $T(n/3) \leq c \cdot \log_3(n/3)$ is ture for some c.

$$
\begin{aligned}
T(n) &= T(n/3) + 1 \\
&\leq c \cdot \log_3(n/3) + 1 \\
&= c \log_3 n - c \log_3 3 + 1 \\
&= c \log_3 n - c + 1 \\
&\leq c \log_3 n \qquad\qquad \text{if } c \geq 1
\end{aligned}
$$

Q6: (10 points)

Use the recursion tree method to solve $T(n) = T(n/3) + 1$.

- The figure on the next page shows the tree, where $T(n) = \log_3 n + 1 = \Theta(\log n)$.

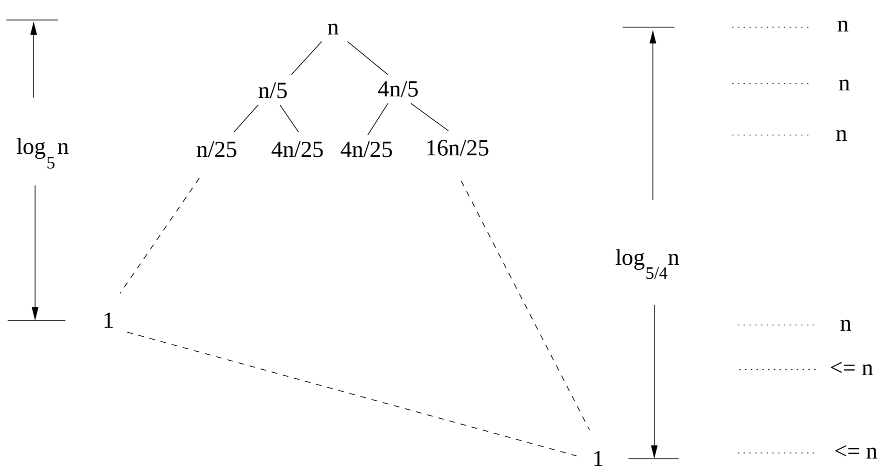


Q7: (10 points)

Please use the recursion tree method to solve the following recurrence

$$T(n) = T(n/5) + T(4n/5) + n$$

- The figure below shows the tree, where $T(n) \geq n \log_5 n = \Omega(n \log n)$ and $T(n) \leq n \log_{5/4} n = O(n \log n)$. Thus, $T(n) = \Theta(n \log n)$.



Q8: Exercise 4.3-1 (10 points)

Use the master method to give tight asymptotic bounds for the following recurrences.

- **(a)** $T(n) = 4T(n/2) + n$
  $n^{\log_b a} = n^{\log_2 4} = n^2$ and $f(n) = n$
  There exist $\epsilon > 0$ such that $f(n) = O(n^{\log_b a - \epsilon})$
  Case 1, $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$

- **(b)** $T(n) = 4T(n/2) + n^2$
  $n^{\log_b a} = n^{\log_2 4} = n^2$ and $f(n) = n^2$
  Since $f(n) = \Theta(n^{\log_b a})$
  Case 2, $T(n) = \Theta(n^{\log_b a} \log n) = \Theta(n^2 \log n)$

- **(c)** $T(n) = 4T(n/2) + n^3$
  $n^{\log_b a} = n^{\log_2 4} = n^2$ and $f(n) = n^3$
  There exists $\epsilon > 0$ such that $f(n) = \Omega(n^{\log_b a + \epsilon})$
  There exists constant $1/2 \leq c \leq 1$ such that
  $a \cdot f(n/b) = 4 \cdot f(n/2) = n^3/2 \leq c \cdot f(n) = c \cdot n^3$ is true.
  Case 3, $T(n) = \Theta(f(n)) = \Theta(n^3)$

Q9: (10 points)

Can the master method to be applied to the recurrence $T(n) = 4T(n/2) + n^2 \log n$? Justify your answer.

- $a = 4, b = 2, f(n) = n^2 \lg n, n^{\lg_b a} = n^2$
  The ratio of $f(n)/n^{\lg_b a} = \lg n$. This implies f(n) is asymptotically larger than $n^{\lg_b a}$, but not polynomially larger.
  Thus, the master method `can not` be applied to solve this recurrence.