CIS507: Design & Analysis of Algorithms

Quiz 1, Spring 2014

Version with answers

Duration: 15 minutes
Total weight: 5%

Student Name: –

Student ID: –

Problem	Points Obtained	Points Possible
1		4
2		1
Total		5

Cheat Sheet: Master Method

For $T(n) = aT(n/b) + f(n)$, with $a \geq 1$, $b > 1$, compare $f(n)$ with $n^{\log_b a}$.

Case	Condition, for $\epsilon > 0$	Solution
1	$f(n) = \mathcal{O}(n^{\log_b a - \epsilon})$	$T(n) = \Theta(n^{\log_b a})$ Number of leafs dominates
2	$f(n) = \Theta(n^{\log_b a})$	$T(n) = \Theta(n^{\log_b a} \log n)$ All rows have same asymptotic sum
3	$f(n) = \Omega(n^{\log_b a + \epsilon})$	$T(n) = \Theta(f(n))$ provided that $af(n/b) \leq cf(n)$ for some $c < 1$
2 (general)	$f(n) = \Theta(n^{\log_b a} \log^k n)$ or some constant $k \geq 0$	$T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$ They grow at 'similar' rate

1 Multiple Choice (4 points)

For each of the following, circle the correct answer(s). Note that there may be more than one in each question.

1. **(0.5 point)** $f(n) = (n^8 + 5)^{0.5}$, and $g(n) = n^4$.
   $f = \mathcal{O}(g)$ $f = \Theta(g)$ $f = \Omega(g^2)$ $f = o(g)$ $f = \omega(g)$

2. **(0.5 point)** Let $f(n) = \log(\sqrt{n})$ and $g(n) = \mathcal{O}(\log n)$.
   $f = \mathcal{O}(\sqrt{g})$ $f = \Theta(g)$ $f = \Omega(g)$ $f = o(g)$ $f = \omega(g)$

3. **(0.5 point)** Let $f(n) = n^{\log n}$, and $g(n) = n^{2 \log n}$.
   $f = \mathcal{O}(g)$ $f = \Theta(g^2)$ $f = \Omega(g)$ $f = o(\log g)$ $f = \omega(g)$

4. **(0.5 point)** $f(n) = 2^{\sqrt{n}}$, and $g(n) = \sqrt{2^n}$.
   $f = \mathcal{O}(g)$ $f = \Theta(g)$ $f = \Omega(g)$ $f = o(g)$ $f = \omega(g)$

5. **(0.5 point)** $f(n) = \sum_{i=0}^{\infty} \left(\frac{1}{n}\right)^i$, and $g(n) = \frac{1}{\log n}$.
   $f = \mathcal{O}(g)$ $f = \Theta(g)$ $f = \Omega(g)$ $f = o(g)$ $f = \omega(g)$

6. **(0.5 point)** $f(n) = (\log(n!))^2$, and $g(n) = n^2 \log n$.
   $f = \mathcal{O}(g)$ $f = \Theta(g)$ $f = \Theta(g \log g)$ $f = o(g)$ $f = \omega(g)$

7. **(0.5 point)** Suppose the running time of an algorithm follows the recurrence $T(n) = 2T(n-1) + \Theta(1)$, and $T(n) = \Theta(1)$ for $n \leq 10$. What is the asymptotic running time?
   $\Theta(n^2)$ $\Theta(2^n)$ $\Theta(\log n)$ $\Theta(1)$

8. **(0.5 point)** Suppose the running time of an algorithm follows the recurrence $T(n) = 2T(\frac{n}{4}) + T(\frac{n}{2}) + \Theta(n)$, and $T(n) = \Theta(1)$ for $n \leq 10$. What is the asymptotic running time?
   $\Theta(n)$ $\Theta(n^{0.75})$ $\Theta(\log n)$ $\Theta(n \log n)$

ANSWER:

1. $f = O(g)$; $f = \Theta(g)$.
2. $f = \Omega(g)$.
3. $f = O(g)$.
4. $f = O(g)$; $f = o(g)$.
5. $f = \Omega(g)$; $f = \omega(g)$.
6. $f = \Theta(g \log g)$; $f = \omega(g)$.

7. $T(n) = \Theta(2^n)$.
8. $T(n) = \Theta(n \log n)$.

2 Recurrences (1.0 point)

Give asymptotic upper and lower bounds (using Θ-notation) for $T(n)$ for each of the following recurrences. Assume that $T(n)$ is a non-negative constant for $n \leq 10$. Justify your answers. (*Hint:* Use the Master Method).

1. **(0.5 point)** $T(n) = 8T(n/2) + (n \log n)^3$
2. **(0.5 point)** $T(n) = T(n/7) + \Theta(1)$

ANSWER:

1. $f(n) = \Theta(n^{\log_2 8} \log^3 n)$. Case 2 (general): $T(n) = \Theta(n^3 \log^4 n)$.
2. $f(n) = \Theta(n^{\log_7 1})$. Case 2: $T(n) = \Theta(\log n)$.