Homework 2 Solutions

1.

① Function order of Growth (20 pts)

List the 4 functions below in non-decreasing asymptotic order of growth.

20

$(\log n)^2$ $\quad n^{-2}$ $\quad \lg(2^{\log(n^2)})$ $\quad n^2$

Justify your answer mathematically by showing values of c and n_0 for each pair of functions that are adjacent in your ordering.

Increasing order:	1. n^{-2}	by Justification 1
	2. $\lg(2^{\log(n^2)})$	by Justification 1
	3. $(\log n)^2$	by Justification 2
	4. n^2	by Justification 3

By $\log_b a^n = n \log_b a$

$$\log(2^{\log n^2}) = (\log n^2)\cdot(\log 2) \quad (\log 2 = 1)$$

$$= \log n^2$$

$$= 2 \log n \qquad \text{by } \log_b a^n = n \log_b a$$

And $(\log n)^2 = \log^2 n$ $\qquad$ by $\log^k n = (\log n)^k$

Justifications:

Using $O(g(n)) = \{ f(n) :$ there exist positive constants c and n_0 such that $0 \le f(n) \le c\,g(n)$ for all $n \ge n_0 \}$

1. $O(\lg(2^{\log(n^2)}) = O(2\log n)$ and $O(n^{-2}) = O(2 \log n)$:

Choose $c = 1$, $n_0 = 2$

$$0 \le \frac{1}{n^2} \le c(2\log n)$$

$$= 0 \le \frac{1}{4} \le 2\log(2)$$

$$= 0 \le \tfrac{1}{4} \le 2 \qquad \text{which is true.}$$

2. $O(2\log n) = O(\log^2 n)$

Choose $c = 1$, $n_0 = 8$

$$0 \le 2\log n \le c \log^2 n$$

$$= 0 \le 2\log 8 \le \log^2 8$$

$$= 0 \le 6 \le 9 \qquad \text{which is true}$$

3. $O(\log^2 n) = O(n^2)$

Choose $c = 1$, $n_0 = 8$

$$0 \le \log^2 n \le cn^2$$

$$0 \le \log^2 8 \le 8^2 \qquad \text{which is true}$$

$$0 \le 9 \le 64$$

2.

Mystery(n)		
if n is an even number	c_1	1
for i = 1 to n	c_2	$n+1$
for j = n downto n/2	c_3	$(n/2+1)*n \rightarrow n^2/2+n$
print "1"	c_4	$(n/2)*n \rightarrow n^2/2$
else	c_5	
for k = 1 to n/4	c_6	$n/4+1$
for m = 1 to n	c_7	$(n+1)*n/4$
print "2"	c_8	$n*n/4$

$$T(n) = c_1 + c_2(n+1) + c_3(n^2/2+n) + c_4(n^2/2)$$

$$c_1 + c_2 n + c_2 + c_3\frac{n^2}{2} + c_3 n + c_4\frac{n^2}{2}$$

$$(c_1+c_2) + (c_2+c_3)n + (c_3+c_4)\frac{n^2}{2}$$

$$a + bn + cn^2 \longrightarrow \Theta(n^2)$$

$$T(n) = c_6\left(\frac{n}{4}+1\right) + c_7\left(\frac{n^2}{4}+\frac{n}{4}\right) + c_8\left(\frac{n^2}{4}\right)$$

$$c_6\frac{n}{4} + c_6 + c_7\frac{n^2}{4} + c_7\frac{n}{4} + c_8\frac{n^2}{4}$$

$$c_6 + (c_6+c_7)\frac{n}{4} + (c_7+c_8)\frac{n^2}{4}$$

$$a + bn + cn^2 \rightarrow \Theta(n^2)$$

$$\boxed{T(n) = \Theta(n^2)}$$

3. a. True, b. False, c. False , d. True, e. True

Let $\alpha = 1, \beta = e^{\epsilon}$. By a change of variable $x = \log n$, we can see that

$$\log n = o(e^{\epsilon \log n}) = o(n^{\epsilon})$$

Similarly, we have

$$\log\log n = o((\log n)^{\epsilon}) \quad \text{for any } \epsilon > 0$$

$$e^n = o(\lambda^{e^n}) \quad \text{for any } \lambda > 1$$

4. 1

(4) Give an expression for the runtime if the recurrence can be solved w/ the Master Theorem.

(1) $T(n) = 3^n T(\frac{n}{3}) + n^3$

$\boxed{\text{Master Theorem does not apply.}}$

In $T(n) = a T(n/b) + f(n)$

Since $a = 3^n$ and by definition, $a \geq 1$ and is a constant, 3^n is not a constant, and thus doesn't satisfy the requirements.

(2) $T(n) = 5T(\frac{n}{2}) + \sqrt{10}\, n^3$

$a = 5$, $b = 2$, $c = 3$, $f(n) = \sqrt{10}\, n^3$

$\log_b a = \log_2 5 \approx 2.322$

~~$\log_b a < c$ (Case 3)~~

~~$2.322 < 3$ (true)~~

$f(n) = \Omega(n^{\log_b a + \epsilon}) = n^{\log_2 5 + \epsilon}$ where $\epsilon \approx 0.68$ and $\epsilon < 1$

Check if regularity condition holds:

$a f(n/b) = 5(\frac{n^3}{8}) \leq \frac{5}{8} n^3 = c f(n)$ which is true

Thus, $\boxed{T(n) = \Theta(f(n)) = \Theta(n^3)}$ by Case 3.

(3) $T(n) = \frac{1}{4} T(\frac{n}{4}) + n \log n$

$a = \frac{1}{4}$ ~~and since by definition~~, $a \geq 1$ is a constant

$\boxed{\text{~~Master Thm does not apply.~~}}$

~~(4) $T(n) = T(n-1) + 2n$~~

By definition, ~~$b > 1$ is a constant,~~ ~~in this recurrence does not hold.~~

$\boxed{\text{Master Thm does not apply.}}$

(5) $T(n) = 16T(\frac{n}{4}) + n^2$

$a = 16$, $b = 4$, $c = 2$, $f(n) = n^2$

$\log_b a = \log_4 16 = 2$

$\log_b a = c$ (case 2)

~~$f(n) = \Theta(n^c \log^k n)$, $c = 2$, $k = 0$~~

Thus, $T(n) = \Theta(n^{\log_b a} \log^{k+1} n) = \boxed{\Theta(n^2 \log n)}$ by case 2.

5.

Rate of increase in number of subproblems in each recursion = 2 Rate of decrease in subproblem size = 1 with 1 less input

Hence in each level of the tree, there are 2^i nodes each of cost 1 at depth $i = 0, 1, 2, \ldots, n$.

Hence, total cost of the tree is:

$$\begin{aligned} T(n) &= \sum_{i=0}^{n} 2^i \cdot 1 \\ &= \frac{2^{n+1} - 1}{2 - 1} \\ &= 2^{n+1} - 1 \\ &= 2.2^n - 1 \\ &\leq c2^n \\ &= O(2^n) \end{aligned}$$

The last step holds as long as $c \geq 2$ and $n \geq 1$.

Now we have to show that our guess $T(n) = O(2^n)$ holds using the substitution method. Let's refine our guess to $T(n) \leq d(2^n - b)$ for positive constants b and d.

$$\begin{aligned} T(n) &= 2T(n-1) + 1 \\ &\leq 2d \cdot (2^{n-1} - b) + 1 \\ &= d2^n - 2db + 1 \\ &= d2^n - db - (db - 1) \\ &\leq d(2^n - b) \end{aligned}$$

The last step holds as long as $db - 1 \geq 0$.