ECE608, Homework #3 Solution

(1) CLR 4.3-9

$T(n) = 3T(\sqrt{n}) + \lg n.$

Change variables to $m = \lg n \Leftrightarrow n = 2^m$.

$T(2^m) = 3T(2^{m/2}) + m.$

Change functions to $S(m) = T(2^m)$.

$S(m) = 3S(\frac{m}{2}) + m.$

This is solvable by case 1 on the Master Theorem, since $a = 3, b = 2, f(m) = m, m^{\log_2 3}$, and $f(m) = m = O(m^{\lg 3 - \epsilon})$ for $0 < \epsilon < 1$.

Thus, $S(m) = \Theta(m^{\lg 3})$.

$$\begin{aligned} S(m) = \Theta(m) \;\; &\Rightarrow \;\; T(n) = T(2^m) = S(m) = \Theta(m^{\lg 3}) = \Theta((\lg n)^{\lg 3}) \\ &\Rightarrow \;\; T(n) = \Theta((\lg n)^{\lg 3}) \end{aligned}$$

(2) CLR 4.4-8

Because $a \geq 1$ is a constant, $T(a) = O(1)$.

$$\begin{aligned} T(n) &= n + T(a) + T(n-a) \\ &= n + T(a) + ((n-a) + T(a) + T((n-a)-a) \\ &= n + T(a) + (n-a) + T(a) + T(n-2a) \\ &= n + T(a) + (n-a) + T(a) + ((n-2a) + T(a) + T(n-3a)) \\ &= n + T(a) + (n-a) + T(a) + (n-2a) + T(a) + ((n-3a) + T(a) + T(n-4a)) \\ &\vdots \\ &= n + T(a) + (n-a) + T(a) + (n-2a) + T(a) + \ldots + (n-ia) + T(a) + T(n-(i+1)a) \\ &\vdots \\ &= \sum_{i=0}^{\frac{n-2a}{a}-1} ((n-ia) + T(a)) + T(n - a\left(\frac{n-2a}{a} + 1\right)) \\ &= \sum_{i=0}^{\frac{n-2a}{a}-1} ((n-ia) + T(a)) + T(a) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=0}^{\frac{n-2a}{a}-1}(n-ia) + \sum_{i=0}^{\frac{n-2a}{a}} T(a) \\
&= n\sum_{i=0}^{\frac{n-2a}{a}-1} 1 - a\sum_{i=0}^{\frac{n-2a}{a}-1} i + T(a)\sum_{i=0}^{\frac{n-2a}{a}} 1 \\
&= n(\frac{n-2a}{a}) - a\sum_{i=1}^{\frac{n-2a}{a}-1} i + \Theta(1)(\frac{n-2a}{a}+1) \\
&= n(\frac{n-2a}{a}) - a(\frac{\frac{n-2a}{a}(\frac{n-2a}{a}-1)}{2}) + \Theta(n) \\
&= n(\frac{n-2a}{a}) - a(\frac{\frac{n-2a}{a}(\frac{n-2a}{a}-1)}{2}) + \Theta(n) \\
&= \frac{2n\,(n-2a) - a^2(\frac{n-2a}{a}(\frac{n-2a-a}{a}))}{2a} + \Theta(n) \\
&= \frac{2n\,(n-2a) - ((n-2a)(n-3a))}{2a} + \Theta(n) \\
&= \frac{(n-2a)(2n-n+3a)}{2a} + \Theta(n) \\
&= \frac{(n-2a)(n+3a)}{2a} + \Theta(n) = \Theta(n^2)
\end{aligned}
$$

(3) CLR 4.4-9

Consider three cases:

1. $\alpha = 1 - \alpha = 0.5$
2. $\alpha > 1 - \alpha$
3. $\alpha < 1 - \alpha$

The first case gives the recurrence $T(n) = 2T(\frac{n}{2}) + cn$ and the master theorem can be applied. Here $a = 2$, $b = 2$, and $f(n) = cn$. Since $f(n) = \Theta(n^{\log_b a})$, case 2 can be applied and $T(n) = \Theta(n \log n)$. Since cases 2 and 3 are symmetric, assume without loss of generality that $0 < \alpha < \frac{1}{2}$ and thus $\frac{1}{2} < 1 - \alpha < 1$ (see Figure 1).

Since $\alpha < \frac{1}{2}$, the shortest path from root to leaf is $n \to \alpha n \to \alpha^2 n \to \ldots \to \alpha^k n \to \ldots \to 1$. The minimum height of the tree is obtained as follows:

$$\alpha^k n = 1 \Rightarrow n = \frac{1}{\alpha^k} \Rightarrow k = \log_{\frac{1}{\alpha}} n$$

Thus, we can informally calculate a lower bound on the tree as:

$$T(n) \;=\; \sum_{i=0}^{\log_{\frac{1}{\alpha}} n-1} n + 2^{\log_{\frac{1}{\alpha}} n}$$

$$
\begin{aligned}
&\geq \sum_{i=0}^{\log_{\frac{1}{\alpha}} n - 1} n \\
&= n \log_{\frac{1}{\alpha}} n \\
&= n \frac{\lg n}{\lg \frac{1}{\alpha}} \\
&\geq c\, n \lg n \qquad \triangleright\, c = \frac{1}{\lg \frac{1}{\alpha}} \\
&= \Omega(n \lg n)
\end{aligned}
$$

Let's prove that $T(n) = \Omega(n \lg n)$ is correct using substitution, namely that $T(n) \geq d\, n \lg n$, where d is a suitable constant.

Base Case: $T(2) = T(\alpha 2) + T((1-\alpha)2) + 2c = O(1) + O(1) + 2c \geq d\, 2 \lg 2 = 2d$, take $1 \leq d < c$.

Induction Step: Assume for all $k \leq n$ that $T(k) \geq d\, k \lg k$.

$$
\begin{aligned}
T(n) &= T(\alpha n) + T((1-\alpha)n) + cn \\
&\geq d\, \alpha n \lg(\alpha n) + d\,(1-\alpha) n \lg((1-\alpha)n) + cn \\
&= d\, \alpha n(\lg \alpha + \lg n) + d\,(1-\alpha)n(\lg(1-\alpha) + \lg n) + cn \\
&= d\, \alpha n \lg n + d\, \alpha n \lg \alpha + d\,(1-\alpha) n \lg n + d\,(1-\alpha) n \lg(1-\alpha) + cn \\
&= d\, n \lg n(\alpha + (1-\alpha)) + d\, \alpha n \lg \alpha + d\,(1-\alpha) n \lg(1-\alpha) + cn \\
&= d\, n \lg n + d\, \alpha n \lg \alpha + d\,(1-\alpha) n \lg(1-\alpha) + cn \\
&= d\, n \lg n - d\, n(\alpha \lg \frac{1}{\alpha} + (1-\alpha) \lg \frac{1}{1-\alpha}) + cn \\
&= d\, n \lg n + n(c - d\,(\alpha \lg \frac{1}{\alpha} + (1-\alpha) \lg \frac{1}{1-\alpha})) \\
&\geq d\, n \lg n
\end{aligned}
$$

Assuming that $c \geq d\,(\alpha \lg \frac{1}{\alpha} + (1-\alpha) \lg \frac{1}{1-\alpha})$.

On the other hand, the longest path from root to leaf is $n \to (1-\alpha)n \to (1-\alpha)^2 n \to \ldots \to (1-\alpha)^k n \to \ldots \to 1$. The greatest height of the tree is calculated as follows:

$$
(1-\alpha)^k n = 1 \Rightarrow n = \frac{1}{(1-\alpha)^k} \Rightarrow k = \log_{\frac{1}{1-\alpha}} n
$$

Thus, we can informally calculate a upper bound on the tree as follows, since there will be a single leaf at the greatest depth given our recurrence.

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log_{\frac{1}{1-\alpha}} n-1} n + O(1) \\
&\leq \sum_{i=0}^{\log_{\frac{1}{1-\alpha}} n-1} n + a \\
&= n \log_{\frac{1}{1-\alpha}} n + a \\
&= n \frac{\lg n}{\lg \frac{1}{1-\alpha}} + a \\
&\leq n \lg n + a \\
&\leq c\, n \lg n \qquad \rhd\, c \text{ so that } c\, n \lg n > n \lg n + a \\
&= O(n \lg n)
\end{aligned}
$$

Let's prove that $T(n) = O(n \lg n)$ is correct using substitution, namely that $T(n) \leq d\, n \lg n$, where d is a suitable constant.

Base Case: $T(2) = T(\alpha 2) + T((1-\alpha)2) + 2c = O(1) + O(1) + 2c \leq d\, 2 \lg 2 = 2d$, take $1 <= c < d$.

Induction Step: Assume for all $k \leq n$ that $T(k) \leq d\, k \lg k$.

$$
\begin{aligned}
T(n) &= T(\alpha n) + T((1-\alpha)n) + cn \\
&\leq d\, \alpha n \lg(\alpha n) + d\,(1-\alpha)n \lg((1-\alpha)n) + cn \\
&= d\, \alpha n(\lg \alpha + \lg n) + d\,(1-\alpha)n(\lg(1-\alpha) + \lg n) + cn \\
&= d\, \alpha n \lg n + d\, \alpha n \lg \alpha + d\,(1-\alpha)n \lg n + d\,(1-\alpha)n \lg(1-\alpha) + cn \\
&= d\, n \lg n(\alpha + (1-\alpha)) + d\, \alpha n \lg \alpha + d\,(1-\alpha)n \lg(1-\alpha) + cn \\
&= d\, n \lg n + d\, \alpha n \lg \alpha + d\,(1-\alpha)n \lg(1-\alpha) + cn \\
&= d\, n \lg n - d\, n(\alpha \lg \frac{1}{\alpha} + (1-\alpha) \lg \frac{1}{1-\alpha}) + cn \\
&= d\, n \lg n + n(c - d\,(\alpha \lg \frac{1}{\alpha} + (1-\alpha) \lg \frac{1}{1-\alpha})) \\
&\leq d\, n \lg n
\end{aligned}
$$

Assuming that $c \leq d\,(\alpha \lg \frac{1}{\alpha} + (1-\alpha) \lg \frac{1}{1-\alpha})$.
Therefore, $T(n) = \Theta(n \lg n)$.

(4) CLR 4.5-1

1. $T(n) = 2T(\frac{n}{4}) + 1$

   $n^{\log_b a} = n^{\log_4 2} = \sqrt{n}$, and $f(n) = 1$. Because $f(n) = 1 = O(n^{\frac{1}{2-\epsilon}})$ for $0 < \epsilon \leq 1$, Case 1 of the Master Theorem applies, and so $T(n) = \Theta(\sqrt{n})$.

2. $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$

   $n^{\log_b a} = n^{\log_4 2} = \sqrt{n}$, and $f(n) = \sqrt{n}$. Because $f(n) = \sqrt{n} = n^{\log_b a}$, Case 2 of the Master Theorem applies, and so $T(n) = \Theta(\sqrt{n}\lg n)$.

3. $T(n) = 2T(\frac{n}{4}) + n$

   $n^{\log_b a} = n^{\log_4 2} = \sqrt{n}$, and $f(n) = n$. Because $f(n) = n = \Omega(n^{\frac{1}{2+\epsilon}})$ for $0 < \epsilon < 1$ and the regularity condition holds: $a\, f(\frac{n}{b}) \leq c\, f(n)$ for some $c < 1$, that is, $2\,(\frac{n}{4}) = \frac{n}{2} \leq c\,n$, when $c = \frac{1}{2}$, Case 3 of the Master Theorem applies, and so $T(n) = \Theta(n)$.

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

   $n^{\log_b a} = n^{\log_4 2} = \sqrt{n}$, and $f(n) = n^2$. Because $f(n) = n^2 = \Omega(n^{\frac{1}{2+\epsilon}})$ for $0 < \epsilon < 1$ and the regularity condition holds: $a\, f(\frac{n}{b}) \leq c\, f(n)$ for some $c < 1$, that is, $2\,(\frac{n}{4}) = \frac{n}{2} \leq c\,n$, when $c = \frac{1}{2}$, Case 3 of the Master Theorem applies, and so $T(n) = \Theta(n^2)$.

(5) CLR 4.5-3

Since $T(n) = T(\frac{n}{2}) + \Theta(1)$, we will use the Master theorem.
We have $a = 1, b = 2$, and $f(n) = \Theta(1)$.

$$n^{\log_b a} = n^{\log_2 1} = 1 = \Theta(1)$$

By case 2 of the Master theorem, $T(n) = \Theta(\lg n)$.

(6) CLR 4.5-4

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

For the given recurrence equation $a = 4$ and $b = 2$, and $\log_2 4 = 2$.

We can see that $f(n) = n^2 \lg n \neq O(n^{2-\epsilon})$ for $\epsilon > 0$. Also $n^2 \lg n \neq \Theta(n^2)$, and $n^2 \lg n \neq \Omega(n^{2+\epsilon})$ for $\epsilon > 0$. Thus we see that none of the three cases in the master method are applicable to this recurrence equation.

We can try to asymptotically bound this recurrence using a guess and verifying it using induction. We guess that $T(n) = \Theta(n^2 \lg^2 n)$.

We will now identify c_1 and c_2 so that we can show $c_1 n^2 \lg^2 n \leq T(n) \leq c_2 n^2 \lg n$ by induction, for all $n \geq 2$ to conclude $T(n) = \Theta(n^2 \lg^2 n)$. By definition, $T(2) = 4T(1)+$

$2^2 \lg 2 = 8$, so for the base case we need $c_1 2^2 \lg^2(2) = 4c_1 \leq 8 \leq 4c_2 = c_2 2^2 \lg^2(2)$, requiring that $c_1 \leq 2$ and $c_2 \geq 2$.

For the inductive case, we need the choice of c_1 and c_2 to enable the proof of $c_1 n^2 \lg^2 n \leq T(n) \leq c_2 n^2 \lg^2 n$ from the inductive hypothesis that $c_1 k^2 \lg^2 k \leq T(k) \leq c_2 k^2 \lg^2 k$ for all k such that $2 \leq k < n$.

Applying the definition of $T(n)$ and the inductive hypothesis with $k = \frac{n}{2}$, what we need to show converts to

$$c_1 n^2 \lg^2 n \leq 4c_1 \frac{n^2}{4} \lg^2 \frac{n}{2} \leq T(n)$$

and

$$T(n) \leq 4c_2 \frac{n^2}{4} \lg^2 \frac{n}{2} \leq c_2 n^2 \lg^2 n$$

.

Simple algebra, using $\lg 2 = 1$, verifies that the first of these is satisfied whenever $c_1 \leq \frac{2 \lg n - 1}{\lg n}$ and the second whenever $c_2 \geq \frac{\lg n}{2 \lg n - 1}$. Combining these conditions with the base case restrictions on c_1 and c_2, and the fact that $n \geq 2$, we find that the induction completes successfully for $c_1 = 1$ and $c_2 = 2$.

Thus $T(n) = \Theta(n^2 \lg^2 n)$ as desired.

(7) CLR 4.6-2

We wish to show that if $f(n) = \Theta(n^{\log_b a} \lg^k n)$, for $k \geq 0$, then the solution to the Master recurrence is $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$. We confine our analysis to exact powers of b.

From Lemma 4.2, which assumes exact powers of b, the Master recurrence is expanded into the sum or two terms: $T(n) = \Theta(n^{\log_b a}) + \sum_{i=0}^{\log_b n - 1} a^i f(\frac{n}{b^i})$. Let $g(n) = \sum_{i=0}^{\log_b n - 1} a^i f(\frac{n}{b^i})$.

Now given that $f(n) = \Theta(n^{\log_b a} \lg^k n)$, for $k \geq 0$, we can show that $g(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$ (similar to the proof of case 2 in Lemma 4.3). Because $f(n) = \Theta(n^{\log_b a} \lg^k n)$, we have $f(\frac{n}{b^i}) = \Theta((\frac{n}{b^i})^{\log_b a} \lg^k \frac{n}{b^i})$. Substituting this into the $g(n)$ equation:

$$
\begin{aligned}
g(n) &= \sum_{i=0}^{\log_b n-1} a^i\, f(\tfrac{n}{b^i}) = \sum_{i=0}^{\log_b n-1} a^i\,\Theta((\tfrac{n}{b^i})^{\log_b a}\ \lg^k \tfrac{n}{b^i}) \\
&= \Theta(\sum_{i=0}^{\log_b n-1} a^i\,(\tfrac{n}{b^i})^{\log_b a}\ \lg^k \tfrac{n}{b^i}) \\
&= \Theta(n^{\log_b a} \sum_{i=0}^{\log_b n-1} (\tfrac{a}{b^{\log_b a}})^i\,(\lg \tfrac{n}{b^i})^k) \\
&= \Theta(n^{\log_b a} \sum_{i=0}^{\log_b n-1} 1^i\,(\lg n - i\lg b)^k) \\
&= \Theta(n^{\log_b a} \sum_{i=0}^{\log_b n-1} \lg^k n(1 - i\tfrac{\lg b}{\lg n})^k) \\
&= \Theta(n^{\log_b a} \lg^k n \sum_{i=0}^{\log_b n-1} (1 - i\tfrac{\lg b}{\lg n})^k)
\end{aligned}
$$

The summation is bounded by $\Theta(\log_b n) = \Theta(\lg n)$ as we show shortly below. Hence:

$$
\begin{aligned}
&= \Theta(n^{\log_b a} \lg^k n \lg n) \\
&= \Theta(n^{\log_b a} \lg^{k+1} n)
\end{aligned}
$$

All that remains is to show that $\displaystyle\sum_{i=0}^{\log_b n-1} (1-i\tfrac{\lg b}{\lg n})^k = \Theta(\log_b n)$. First, the lower bound:

$$
\begin{aligned}
\sum_{i=0}^{\log_b n-1} (1 - i\tfrac{\lg b}{\lg n})^k &\geq \sum_{i=0}^{\lfloor\frac{\log_b n-1}{2}\rfloor} (1 - \tfrac{\log_b n}{2}\tfrac{\lg b}{\lg n})^k \ + \sum_{i=\lfloor\frac{\log_b n-1}{2}\rfloor+1}^{\log_b n-1} (1 - \log_b n\tfrac{\lg b}{\lg n})^k \\
&\geq \sum_{i=0}^{\lfloor\frac{\log_b n-1}{2}\rfloor} (1 - \tfrac{1}{2}\tfrac{\lg n}{\lg b}\tfrac{\lg b}{\lg n})^k \ + \sum_{i=\lfloor\frac{\log_b n-1}{2}\rfloor+1}^{\log_b n-1} (1 - \tfrac{\lg n}{\lg b}\tfrac{\lg b}{\lg n})^k \\
&= \sum_{i=0}^{\lfloor\frac{\log_b n-1}{2}\rfloor} (1 - \tfrac{1}{2})^k \ + 0 = \sum_{i=0}^{\lfloor\frac{\log_b n-1}{2}\rfloor} (\tfrac{1}{2})^k \\
&= \Omega(\frac{\log_b n - 1}{2}(\tfrac{1}{2})^k) = \Omega(\log_b n(\tfrac{1}{2})^{k+1}) \ = \Omega(\log_b n)
\end{aligned}
$$

Next, the upper bound:

$$\sum_{i=0}^{\log_b n-1}(1-i\tfrac{\lg b}{\lg n})^k \le \sum_{i=0}^{\log_b n-1}(1-0\tfrac{\lg b}{\lg n})^k = \sum_{i=0}^{\log_b n-1}1^k$$
$$= O(\log_b n)$$

Hence, $T(n) = \Theta(n^{\log_b a}) + \Theta(n^{\log_b a}(\lg^{k+1} n)) = \Theta(n^{\log_b a}(\lg^{k+1} n))$.

(8) CLR 4-1

In parts (a), (b), and (d), we are applying case 3 of the master theorem, which requires that $a\mathrm{f}(\frac{n}{b}) \le c\mathrm{f}(n)$ for $c < 1$. In each of these parts, f(n) has the form n^i. The desired condition is satisfied because $a\mathrm{f}(\frac{n}{b}) = a\frac{n^i}{b^i} = (\frac{a}{b^i})n^i = (\frac{a}{b^i})\mathrm{f}(n)$ and in each of the cases below, $(\frac{a}{b^i}) < 1$ (e.g., in part (a), $\frac{a}{b^i} = \frac{a}{b^3} = \frac{2}{8} < 1$).

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

This is a divide-and-conquer recurrence with $a = 2$, $b = 2$, $f(n) = n^4$, and $n^{\log_b a} = n$. The ratio $\frac{f(n)}{n^{\log_b a}} = \frac{n^4}{n} = n^3$ (or $f(n) = n^4 = \Omega(n^{2+\epsilon})$), $0 < \epsilon \le 1$), and $2(\frac{n}{2})^4 = \frac{n^4}{8} \le c\,n^4$ for $\frac{1}{8} \le c < 1$. Case 3 of the master theorem applies, so $\mathrm{T}(n) = \Theta(n^4)$.

(b) $T(n) = T(\frac{7n}{10}) + n$

This is a divide-and-conquer recurrence with $a = 1$, $b = \frac{10}{7}$, $f(n) = n$, and $n^{\log_b a} = n^0 = 1$. The ratio $\frac{f(n)}{n^{\log_b a}} = \frac{n}{1} = n$ (or $f(n) = n = \Omega(n^{0+\epsilon})$, $0 < \epsilon \le 1$), and $\frac{7n}{10} \le c\,n$ for $\frac{7}{10} \le c < 1$. Case 3 of the master theorem applies, so $\mathrm{T}(n) = \Theta(n)$.

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

This is a divide-and-conquer recurrence with $a = 16$, $b = 4$, $f(n) = n^2$, and $n^{\log_b a} = n^2$. The ratio $\frac{f(n)}{n^{\log_b a}} = \frac{n^2}{n^2} = 1$ (or $f(n) = n^2 = \Theta(n^2)$). Case 2 of the master theorem applies, so $T(n) = \Theta(n^2 \lg n)$.

(d) $T(n) = 7T(\frac{n}{3}) + n^2$

This is a divide-and-conquer recurrence with $a = 7$, $b = 3$, $f(n) = n^2$, and $n^{\log_b a} = n^{\log_3 7}$. The ratio $\frac{f(n)}{n^{\log_b a}} = \frac{n^2}{n^{\log_3 7}} = n^{2-\log_3 7}$ (or $f(n) = n^2 = \Omega(n^{\log_3 7+\epsilon})$, $\log_3 7 < 2$), and $7\,(\frac{n}{3})^2 = \frac{7}{9}\,n^2 \le c\,n^2$, for $\frac{7}{9} \le c < 1$. We know that $2 - \log_3 7 > 0$, since $1 < \log_3 7 < 2$, so case 3 of the master theorem applies, so $T(n) = \Theta(n^2)$.

(e) $T(n) = 7T(\frac{n}{2}) + n^2$

This is a divide-and-conquer recurrence with $a = 7$, $b = 2$, $f(n) = n^2$, and $n^{\log_b a} = n^{\log_2 7}$. The ratio $\frac{f(n)}{n^{\log_b a}} = \frac{n^2}{n^{\log_2 7}} = n^{2-\log_2 7}$ (or $f(n) = n^2 = O(n^{\log_2 7-\epsilon})$,

$2 < \log_2 7$). We know that $2 - \log_2 7 < 0$, since $2 < \log_2 7 < 3$, so case 1 of the master theorem applies, so $T(n) = \Theta(n^{\lg 7})$.

(f) $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$

This is a divide-and-conquer recurrence with $a = 2$, $b = 4$, $f(n) = \sqrt{n}$, and $n^{\log_b a} = \sqrt{n}$. The ratio $\dfrac{f(n)}{n^{\log_b a}} = \dfrac{\sqrt{n}}{\sqrt{n}} = 1$ (or $f(n) = \Theta(\sqrt{n})$). Case 2 of the master theorem applies, so $T(n) = \Theta(\sqrt{n}\lg n)$.

(g) $T(n) = T(n-2) + n^2$

This recurrence can be solved by the iteration method.

$$\begin{aligned}\mathrm{T}(n) &= n^2 + \mathrm{T}(n-2) = n^2 + (n-2)^2 + \mathrm{T}(n-4)\\ &= n^2 + (n-2)^2 + (n-4)^2 + \cdots + 2^2 + \mathrm{T}(1)\end{aligned}$$

For even n:

$$\mathrm{T}(n) = \sum_{i=2}^{\frac{n}{2}} (2i)^2 + \Theta(1) = \frac{(n(n+2)(n+1))}{6} - 1 + \Theta(1) = \Theta(n^3).$$

For odd n:

$$\mathrm{T}(n) = \sum_{i=2}^{\frac{(n-1)}{2}} (2i-1)^2 + \Theta(1) = \Theta(n^3).$$

If you solve the above summation, you will observe that $T(n) = \Theta(n^3)$.

(h) $T(n) = T(\sqrt{n}) + 1$

This recurrence can be solved by iteration.

$\mathrm{T}(n) = 1 + \mathrm{T}(n^{\frac{1}{2}}) = 1 + 1 + \mathrm{T}(n^{\frac{1}{4}}) = 1 + 1 + \cdots + 1 + \mathrm{T}(2)$

It will iterate $\lceil i \rceil$ times, where $n^{\frac{1}{2^i}} = 2$. Solving for i, $n^{\frac{1}{2^i}} = 2 \Rightarrow (\dfrac{1}{2^i})\lg n = 1$ $\Rightarrow 2^i = \lg n \Rightarrow i = \lg\lg n$. Hence, $\mathrm{T}(n) = \Theta(\lg\lg n) + \Theta(1) = \Theta(\lg\lg n)$.

Here is another solution:

Assume that $n = 2^m$, $\lg n = m$. $T(2^m) = T(2^{\frac{m}{2}}) + 1$, let $S(m) = T(2^m)$, $S(m) = S(\frac{m}{2}) + 1$. Since $n^{\log_2 1} = n^0$, by applying case 2 of master theorem, $S(m) = \Theta(\lg m)$. Hence $T(2^m) = S(m) = \Theta(\lg m)$, $T(n) = \Theta(\lg\lg n)$.

(9) CLR 4-2

The numbers from 0 to n will require $\lg n$ bits to be represented. The values taken on by each bit for the numbers ranging 0 to n follow a particular pattern. Each bit takes on the value 1 and 0 an equal number of times, specifically $\frac{n}{2}$ times. Moreover, for all the array elements with a 1 (or 0) in a particular bit position, each remaining bit position takes the value 1 and 0 an equal number of times.

However, because there is one number missing, every bit position in the contents of the array $A[1..n]$ will contain unequal number of ones and zeros. And we simply need to look at each bit position one at a time to find the missing number. At each bit position we can see whether a 1 or a 0 is missing and that is the value for that particuler bit of the missing number. Note however that because of the observation made in the previous paragraph, at each successive bit position, the number of array elements that need to be examined is halved.

A recursive method to look for this number is as follows: Start with the most significant bit position and scan through all n elements of A, separating the 1's from the 0's. We maintain two pointers, one at the beginning and one at the end of the array. Lets call the pointers head and tail. Starting from head, we scan the array one element at a time until we find a 1 in the highest bit position. Similarily we start from tail and scan the array in reverse order, one element at a time until we find a 0. Swap the two elements that are pointed to by head and tail. Then continue to repeat the same process until head and tail pointers meet. At that point we will have seperated all the numbers with 1's and 0's in that bit position. Simply look at the value of the head pointer, if it is less than $A/2$ there are fewer 0's and the missing number has a zero at the examined bit position. If it is greater than $A/2$ there are fewer 1's and the missing number has a 1 at the examined bit position. Discard that half̈ẅhich is greater, and recursively apply the same procedure to the smaller half for the next bit position. The recursion depth will be the number of bits, and at the end of the n'th recursion the value for each bit of the missing number will be known.

The running time can be expressed as a recursion:

$$T(n) = T(\tfrac{n}{2}) + n$$

Applying the Master Method, we have $a = 1$ and $b = 2$. As $\log_2 1 = 0$, and $\frac{n}{2} \leq cn$, we have from the third case of the Master Method that $T(n) = \Theta(n)$

(10) CLR 4-3

(a) $T(n) = 4T(\frac{n}{3}) + n \lg n$

Apply the Master Theorem, case 1, with $a = 4, b = 3, f(n) = n \lg n$.

$n^{\log_b a} \approx n^{1.262}$

Since $f(n) = O(n^{1.262-\epsilon})$ when $\epsilon = .1$, $T(n) = \Theta(n^{\log_3 4})$.

(b) $T(n) = 3T(\frac{n}{3}) + \frac{n}{\lg n}$.

$a = 3, b = 3, f(n) = \frac{n}{\lg n}$

$n^{\log_b a} = n$

Since $f(n)$ is not $O(n^{1-\epsilon})$, is not $\Omega(n^{1+\epsilon})$, and $\frac{f(n)}{n^{\log_b a}} = \frac{1}{\lg n} \neq \Theta(\lg^k n), k \geq 0$, the Master Theorem cannot be applied.

$$\begin{aligned} T(n) &= \frac{n}{\lg n} + 3T(\frac{n}{3}) \\ &= \frac{n}{\lg n} + 3(\frac{n}{3 \lg \frac{n}{3}} + 3T(\frac{n}{9})) \end{aligned}$$

$$
\begin{aligned}
&= \frac{n}{\lg n} + \frac{n}{\lg \frac{n}{3}} + 9T(\frac{n}{9}) \\
&\vdots \\
&= \sum_{j=0}^{i-1} \frac{n}{\lg(\frac{n}{3^j})} + 3^i T(\frac{n}{3^i}) \rhd \frac{n}{3^i} = 1, \text{so } \log_3 n = i \\
&\vdots \\
&= \sum_{j=0}^{\log_3 n - 1} \frac{n}{\lg(\frac{n}{3^j})} + \Theta(1) 3^{\log_3 n} \\
&= \sum_{j=0}^{\log_3 n - 1} \frac{n}{\log_3(\frac{n}{3^j})} + \Theta(n)
\end{aligned}
$$

Let's work with the sum substituting in 3^i for n :

$$
\begin{aligned}
\sum_{j=0}^{i-1} \frac{3^i}{\log_3(\frac{3^i}{3^j})} &= \sum_{j=0}^{i-1} \frac{3^i}{\log_3(3^i) - \log_3(3^j)} \\
&= 3^i \sum_{j=0}^{i-1} \frac{1}{(i-j)\log_3 3} \\
&= 3^i \sum_{j=0}^{i-1} \frac{1}{i-j} \\
&= 3^i \sum_{u=1}^{i} \frac{1}{u}, \text{where } u = i - j \\
&= 3^i H_i \\
&= 3^i(\ln i + \Theta(1)), \text{ substitute } n \text{ for } 3^i \\
&= n(\ln \log_3 n + \Theta(1)) \\
&= n \ln \log_3 n + \Theta(n) \\
&= \Theta(n \lg \lg n)
\end{aligned}
$$

So,

$$
\begin{aligned}
T(n) &= \sum_{j=0}^{\log_3 n - 1} \frac{n}{\log_3(\frac{n}{3^j})} + \Theta(n) \\
&= \Theta(n \lg \lg n) + \Theta(n) \\
&= \Theta(n \lg \lg n)
\end{aligned}
$$

(c) $T(n) = 4T(\frac{n}{2}) + n^2\sqrt{n}$.

$a = 4, b = 2, f(n) = n^2\sqrt{n} = n^{\frac{5}{2}}$, $n^{\log_b a} = n^{\log_2 4} = n^2$, thus
$f(n) = \Omega(n^{\log_b a + \epsilon})$, where $\frac{1}{2} \geq \epsilon > 0$, and $af(\frac{n}{b}) = 4(\frac{n}{2})^{\frac{5}{2}} = 4(\frac{n^{\frac{5}{2}}}{2^{\frac{5}{2}}}) \leq cn^{\frac{5}{2}} = cf(n)$, for $\frac{4}{2^{\frac{5}{2}}} \leq c < 1$.

From case 3 of the Master Theorem, $T(n) = \Theta(n^{\frac{5}{2}})$.

(d) $T(n) = 3T(\frac{n}{3} - 2) + \frac{n}{2}$

Consider the related problem $S(n) = 3S(\frac{n}{3}) + \frac{n}{2}$.
$a = 3, b = 3, f(n) = \frac{n}{2}$, $n^{\log_b a} = n$
Since $f(n) = \Theta(n), S(n) = \Theta(n \lg n)$ by case 2 of the Master Theorem.
However, the Master Theorem does not apply to $T(n) = 3T(\frac{n}{3} - 2) + \frac{n}{2} = 3T(\frac{n-6}{3}) + \frac{n}{2}$, so we must prove it by substitution, by showing that $T(n) = O(n \lg n)$ and $T(n) = \Omega(n \lg n)$.

$O(n \lg n)$**:**

We will work with the form: $T(n) \leq c_1 \, n \lg n + c_2 \, n$.
Base case : Let $T(i) = 1$ for $1 \leq i \leq 8$.
Then, $T(9) = 3T(\frac{3}{3}) + \frac{9}{2} = 3T(1) + 4.5 = 7.5 \leq c_1 9 \lg 9 + 9 \, c_2 = 28.529 \, c_1 - +9 \, c_2$ for $c_2 = 1, c_1 = 1$.
Assume $T(k) \leq c_1 \, k \lg k + c_2 \, k$ for all $k < n$. Then:

$$
\begin{aligned}
T(n) &= 3T(\frac{n-6}{3}) + \frac{n}{2} \\
&\leq 3(c_1(\frac{n-6}{3}) \lg(\frac{n-6}{3}) - c_2 \frac{n-6}{3}) + \frac{n}{2} \\
&= c_1(n-6) \lg(\frac{n-6}{3}) - c_2(n-6) + \frac{n}{2} \\
&\leq c_1(n-6) \lg(\frac{n}{2}) - c_2 n + 6 \, c_2 + \frac{n}{2}, \\
& \quad \text{since } \frac{n}{2} \geq \frac{n-6}{3} \text{ for } n \geq 12 \\
&= c_1(n-6)(\lg n - \lg 2) - c_2 n + 6 \, c_2 + \frac{n}{2}, \\
&= c_1(n-6)(\lg n - 1) - c_2 n + 6 \, c_2 + \frac{n}{2}, \\
&= c_1(n-6) \lg n - c_1(n-6) - c_2 n + 6 \, c_2 + \frac{n}{2}, \\
&= c_1 \, n \lg n - 6 \, c_1 \lg n - c_1(n-6) - c_2 n + 6 \, c_2 + \frac{n}{2}, \\
&= c_1 \, n \lg n - c_2 \, n - (c_1(n-6) - 6 \, c_2 + 6 \, c_1 \lg n - \frac{n}{2}) \\
. &\leq c_1 n \lg n, \text{ for } c_1 \text{ and } c_2 \text{ such that } c_1(n-6) - 6c_2 + 6c_1 \lg n - \tfrac{n}{2} \geq 0.
\end{aligned}
$$

Therefore, $T(n) = O(n \lg n)$.

$\Omega(n \lg n)$**:**

Base case : Let $T(i) = 1$ for $1 \leq i \leq 8$.
Then, $T(9) = 3T(\frac{3}{3}) + \frac{9}{2} = 3T(1) + 4.5 = 7.5 \geq c9\lg 9 = c\,28.529$ for $c < \frac{1}{4}$.
Assume $T(k) \geq c_1\, k \lg k$, for all $k < n$. Then:

$$\begin{aligned}
T(n) &= 3T(\frac{n-6}{3}) + \frac{n}{2} \\
&\geq 3c(\frac{n-6}{3}) \lg(\frac{n-6}{3}) + \frac{n}{2} \\
&\geq 3c\frac{n-6}{3} \lg(\frac{n}{4}) + \frac{n}{2} \\
&\quad \text{for } n \geq 24 \\
&\geq cn\lg(\frac{n}{4}) - 6c\lg(\frac{n}{4}) + \frac{n}{2} \\
&\geq cn\lg(n) - cn\lg(4) - 6c\lg(n) + 12c + \frac{n}{2} \\
&\geq cn\lg(n) + \frac{n}{2} + 12c - cn\lg(4) - 6c\lg(n) \\
&\geq cn\lg(n), \text{ for } c < \frac{1}{4}
\end{aligned}$$

Recall that $\lg n$ grows slower than n.
Therefore, $T(n) = \Omega(n \lg n)$. Thus, $T(n) = \Theta(n \lg n)$.

(e) $T(n) = 2T(\frac{n}{2}) + \frac{n}{\lg n}$
$a = 2, b = 2, f(n) = \frac{n}{\lg n}$
$n^{\log_b a} = n$
Since $f(n)$ is not $O(n^{1-\epsilon})$, is not $\Omega(n^{1+\epsilon})$, and $\frac{f(n)}{n^{\log_b a}} = \frac{1}{\lg n} \neq \Theta(\lg^k n), k \geq 0$, the Master Theorem cannot be applied.

$$\begin{aligned}
T(n) &= \frac{n}{\lg n} + 2T(\frac{n}{2}) \\
&= \frac{n}{\lg n} + 2(\frac{n}{2\lg\frac{n}{2}} + 2T(\frac{n}{4})) \\
&= \frac{n}{\lg n} + \frac{n}{\lg\frac{n}{2}} + 4T(\frac{n}{4}) \\
&\vdots \\
&= \sum_{j=0}^{i-1} \frac{n}{\lg(\frac{n}{2^j})} + 2^i T(\frac{n}{2^i}) \\
&\vdots \\
&= \sum_{j=0}^{\lg n-1} \frac{n}{\lg(\frac{n}{2^j})} + \Theta(1)2^{\lg n} \rhd \frac{n}{2^i} = 1 \text{ when } \lg n = i
\end{aligned}$$

$$= \sum_{j=0}^{\lg n-1} \frac{n}{\lg(\frac{n}{2^j})} + \Theta(n)$$

Let's work with the sum substituting in 2^i for n :

$$\begin{aligned}
\sum_{j=0}^{i-1} \frac{2^i}{\lg(\frac{2^i}{2^j})} &= \sum_{j=0}^{i-1} \frac{2^i}{\lg(2^i) - \lg(2^j)} \\
&= 2^i \sum_{j=0}^{i-1} \frac{1}{(i-j)\lg 2} \\
&= 2^i \sum_{j=0}^{i-1} \frac{1}{i-j} \\
&= 2^i \sum_{u=1}^{i} \frac{1}{u}, \text{where } u = i - j \\
&= 2^i H_i \\
&= 2^i(\ln i + \Theta(1)), \text{ substitute } n \text{ for } 2^i \\
&= n(\ln \lg n + \Theta(1)) \\
&= n \ln \lg n + \Theta(n) \\
&= \Theta(n \lg \lg n)
\end{aligned}$$

So,

$$\begin{aligned}
T(n) &= \sum_{j=0}^{\lg n-1} \frac{n}{\lg(\frac{n}{2^j})} + \Theta(n) \\
&= \Theta(n \lg \lg n) + \Theta(n) \\
&= \Theta(n \lg \lg n)
\end{aligned}$$

(f) $T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + T(\frac{n}{8}) + n.$

First we show that $T(n) = O(n)$. Assume $T(n) \leq cn$ for some $c > 0$,

$$\begin{aligned}
T(n) &\leq c\frac{n}{2} + c\frac{n}{4} + c\frac{n}{8} + n \\
&= c\frac{7n}{8} + n \\
&= (\frac{7}{8}c + 1)n \\
&\leq cn
\end{aligned}$$

for some $c \geq 8$. Thus $T(n) = O(n)$. Next we show that $T(n) = \Omega(n)$. Assume $T(n) \geq cn$ for some $c > 0$,

$$\begin{aligned} T(n) &\geq c\frac{n}{2} + c\frac{n}{4} + c\frac{n}{8} + n \\ &= c\frac{7n}{8} + n \\ &= (\frac{7}{8}c + 1)n \\ &\geq cn \end{aligned}$$

for some $c \leq 8$. Thus $T(n) = \Omega(n)$.

We have shown that $T(n) = O(n)$ and $T(n) = \Omega(n)$, therefore we can conclude that $T(n) = \Theta(n)$.

(g) $T(n) = T(n-1) + \frac{1}{n}$

$$\begin{aligned} T(n) &= \frac{1}{n} + T(n-1) \\ &= \frac{1}{n} + \frac{1}{n-1} + T(n-2) \\ &\vdots \\ &= \sum_{i=0}^{n-2} \frac{1}{n-i} + T(n-(n-1)) \\ &= \sum_{i=0}^{n-2} \frac{1}{n-i} + T(1) \\ &= \sum_{u=1}^{n} \frac{1}{u} - 1 + \Theta(1), \text{where } u = n - i \\ &= \Theta(\ln n) + \Theta(1) \\ &= \Theta(\ln n) \end{aligned}$$

Note that the nth harmonic number is $\Omega(lnn)$, which can be shown using the summation splitting technique (similar to A.10).

(h) $T(n) = T(n-1) + \lg n$

$$\begin{aligned} T(n) &= \lg n + T(n-1) \\ &= \lg n + \lg(n-1) + T(n-2) \\ &\vdots \\ &= \sum_{i=0}^{n-2} \lg(n-i) + T(n-(n-1)) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=0}^{n-2} \lg(n-i) + T(1) \\
&= \sum_{u=2}^{n} \lg u + \Theta(1), \text{where } u = n-i \\
&= \lg(\prod_{u=2}^{n} u) + \Theta(1) \\
&= \lg(\prod_{u=1}^{n} u) + \Theta(1) \\
&= \lg(n!) + \Theta(1) \\
&= \Theta(n \lg n) + \Theta(1) \\
&= \Theta(n \lg n)
\end{aligned}
$$

(i) $T(n) = T(n-2) + \frac{1}{\lg n}$.

$$
\begin{aligned}
T(n) &= \frac{1}{\lg n} + T(n-2) \\
&= \frac{1}{\lg n} + \frac{1}{\lg(n-2)} + T(n-4) \\
&\vdots \\
&= \sum_{i=0}^{\frac{n}{2}-1} \frac{1}{\lg(n-2i)} + T(2)) \\
&= \sum_{u=2}^{n} \frac{1}{\lg u} + \Theta(1), \text{where } u = \tfrac{n}{2} - i
\end{aligned}
$$

To obtain a lower bound,

$$
\begin{aligned}
T(n) &= \sum_{u=2}^{n} \frac{1}{\lg u} + \Theta(1) \\
&> \sum_{u=2}^{n} \frac{1}{\lg n} \\
&= \frac{1}{\lg n} \sum_{u=2}^{n} 1 \\
&= \frac{1}{\lg n}(n-2) \\
T(n) &= \Omega(\frac{n}{\lg n})
\end{aligned}
$$

To obtain an upper bound,

$$\begin{aligned}
T(n) &= \sum_{u=2}^{n} \frac{1}{\lg u} + \Theta(1) \\
&< \sum_{u=2}^{n} \frac{1}{\lg 2} \\
&= \sum_{u=2}^{n} 1 \\
&= (n-2) \\
\\
T(n) &= O(n)
\end{aligned}$$

(j) $T(n) = \sqrt{n}T(\sqrt{n}) + n$

$$\begin{aligned}
T(n) &= n + n^{1/2}T(n^{1/2}) \\
&= n + n^{1/2}(n^{1/2} + n^{1/4}T(n^{1/4})) \\
&= n + n + n^{3/4}T(n^{1/4}) \\
&= n + n + n^{3/4}(n^{1/4} + n^{1/8}T(n^{1/8})) \\
&= n + n + n + n^{7/8}T(n^{1/8}) \\
&\vdots \\
&= in + n^{\sum_{j=1}^{i} \frac{1}{2^j}} T(n^{\frac{1}{2^i}}) \\
&= in + n^c T(2) \\
&\quad \text{where } n^{\frac{1}{2^i}} = 2 \Rightarrow i = \lg\lg n \\
&\quad \text{and } c = \sum_{j=1}^{i} \frac{1}{2^j} < \sum_{j=0}^{\infty} \frac{1}{2^j} - 1 = \frac{1}{1-\frac{1}{2}} - 1 = 1 \\
&= n \lg\lg n + \Theta(n^c), c \le 1 \\
&= \Theta(n \lg\lg n)
\end{aligned}$$

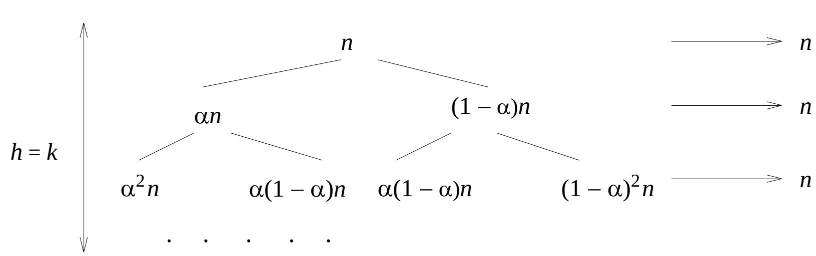

Figure 1: Recurrence Tree for CLR 4.2-5.