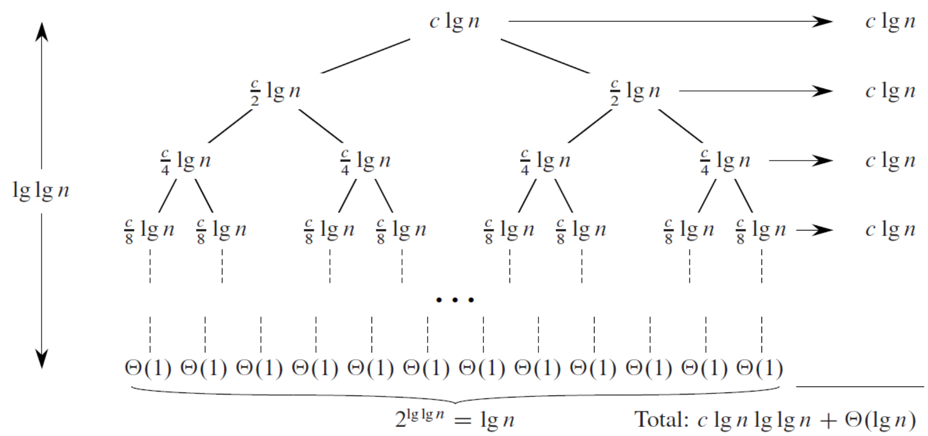
1. Problem 4-3:
   Solution:
   a. Let $m = \log_2 n$, then $n = 2^m$ and $T(n) = T(2^m) = 2T\left(2^{\frac{m}{2}}\right) + \theta(m)$. Define new recurrence $S(m) = T(n)$, then $S(m) = 2S\left(\frac{m}{2}\right) + \theta(m)$.
   b. Use master theorem, we have $S(m) = \theta(m \log_2 m)$.
   c. Substituting $S(m)$ back into $T(n)$, we have $T(n) = \log_2 n\,(\log_2(\log_2 n))$.
   d. Here is the recursion tree:



   The cost of root = $c \log_2 n$. Each of root child =$c \log_2 n^{1/2} = \frac{c}{2} \log_2 n$, so total cost of children's cost = $c \log_2 n$. At the next level down, the cost is the same. Until getting down to a size of 1, we have number of level = $\log_2(\log_2 n)$ and $\log_2 n$ number of leavers ($\theta(1) = c$). Since the problem requires $\log_2 n$ bits to represent n, and number of level = $\log_2(\log_2 n)$. The total cost of all levels above the leaves + total leavers cost = # levels × level cost $+ c \log_2 n$ = $\log_2(\log_2 n) \times c \log_2 n + c \log_2 n = \theta(\log_2 n \log_2(\log_2 n))$.
   e. In order to use Divide-and-Conquer, we need to convert $\sqrt{n}$ to $f(\frac{n}{2})$. Therefore, we let $m = \log_2 n$, then $\sqrt{n} = 2^{m/2}$. Therefore, we define a new recurrence $S(m) = T(n) = 2S\left(\frac{m}{2}\right) + \theta(1)$. Use master theorem, we have $S(m) = \theta(m) = \theta(\log_2 n)$.
   f. Use the same idea of solving the above question, we have $m = \log_3 n$, $S(m) = T(n) = 3S\left(\frac{m}{3}\right) + \theta(3^m)$. Then using the recursive tree (see d), we have # of level = $\log_3 m$, then the total cost = $3 \times 3^{m/3} + 3^2 \times 3^{m/3^2} + 3^3 \times 3^{m/3^3} + \cdots \leq c3^m = \theta(3^m) = \theta(n)$.

2. Problem 4-4:
   Solutions:
   a. We have $f(n) = n \log n$ and $n^{\log_b a} = n^{\log_3 5} \approx n^{1.465}$. Since $n^{\log_3 4-\epsilon}$ for any $0 < \epsilon \leq 0.46$ by case 1 of the master theorem, we have $T(n) = \theta(n^{\log_3 5})$.
   b. According to the recursion tree, $T(n)$ has $\log_3 n$ levels, and number of leaves = $3^{\log_3 n} = n$. Since each leaf cost = $\theta(1)$, the cost of leaves = $\theta(n)$. For the level cost: Total cost of ith level = $\frac{n}{\log_2 \frac{n}{3^i}}$. Thus, $T(n) = \sum_{i=0}^{\log_3 n - 1} \frac{n}{\log_2 \frac{n}{3^i}} + \theta(n)$. Since,

$\log_b a = \frac{\log_c a}{\log_c b}$, $\log_2 \frac{n}{3^i} = \frac{\log_3 \frac{n}{3^i}}{\log_3 2} = \frac{\log_3 n - i}{\log_3 2}$. Thus, $T(n) = \sum_{i=0}^{\log_3 n-1} \frac{n}{\log_2 \frac{n}{3^i}} + \theta(n) = n \log_3 2 \times \left(\sum_{i=0}^{\log_3 n-1} \frac{1}{\log_3 n - i}\right) + \theta(n) = n \log_3 2 \times \left(\sum_{i=0}^{\log_3 n-1} \frac{1}{i}\right) + \theta(n)$. Since $\sum_{i=0}^{n} \frac{1}{i} = \ln n + O(1)$, $T(n) = n \log_3 2 \ln \log_3 n - 1 + \theta(n) = \theta(n \log_3 \log_3 n)$.

c. We have $f(n) = n^3\sqrt{n} = n^{7/2}$ and $n^{\log_b a} = n^{\log_2 8}$. Since $n^{7/2} = \Omega(n^{3+\epsilon})$ for $\epsilon = 1/2$, $f\left(\frac{n}{b}\right) = 8(n/2)^3\sqrt{n/2} = \frac{n^{7/2}}{\sqrt{2}} \le cn^{7/2}, if \frac{1}{\sqrt{2}} \le c < 1$. Applying case 3 of master theorem, we have $T(n) = \theta(n^3\sqrt{n})$.

d. Since $2T\left(\frac{n}{2} - 2\right) + \frac{n}{2}$ is similar to $2T\left(\frac{n}{2}\right) + n$, we guess $T(n) = \theta(n \log n)$.

Proof:

1. Upper bound:

   Since $2T\left(\frac{n}{2} - 2\right) + \frac{n}{2} \le 2T\left(\frac{n}{2}\right) + \frac{n}{2} \le 2T\left(\frac{n}{2}\right) + n$, $T(n) = O((n \log n)$

2. Lower bound:

   Assume, $T(n) \ge cn \log n$, we have: $T(n) \ge 2c\left(\frac{n}{2} - 2\right)\log\left(\frac{n}{2} - 2\right) + \frac{n}{2} = n(cn \log\left(\frac{n}{2} - 2\right) - 4c \log\left(\frac{n}{2} - 2\right) + \frac{n}{2} \ge cn \log\left(\frac{n}{4}\right) - 4c \log\left(\frac{n}{2}\right) + \frac{n}{2}, where\ n \ge 8$. So $T(n) \ge cn \log n - 2cn - 4c \log n + 4c + \frac{n}{2} \ge cn \log n$.

Therefore, $T(n) = \theta(n \log n)$

e. With the same idea that solve the sub-question b and replace $\log_3$ to $\log_2$, $\frac{n}{3^i}$ to $\frac{n}{2^i}$, we can guess $T(n) = \theta(n\log_2\log_2 n)$.

Proof:

1. Upper bound: $T(n) \le n(1 + H_{[\log n]})$, we have: $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n} \le 2\left(\left(\frac{n}{2}\right)\left(1 + H_{\left[\log\frac{n}{2}\right]}\right)\right) + \frac{n}{\log n} = n\left(1 + H_{[\log n-1]}\right) + \frac{n}{\log n} \le n(1 + H_{[\log n]-1} + \frac{1}{\log n} \le n\left(1 + H_{[\log n]-1} + \frac{1}{\log n}\right).since\ H_k = H_{k-1} + \frac{1}{k},\ T(n) \le n(1 + H_{[\log n]})$.

2. Lower bound: $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n} \ge 2\left(\left(\frac{n}{2}\right)\left(H_{\left[\log\frac{n}{2}\right]}\right)\right) + \frac{n}{\log n} = n\left(H_{[\log n-1]}\right) + \frac{n}{\log n} \ge n(H_{[\log n]-1} + \frac{1}{\log n} \ge n\left(H_{[\log n]-1} + \frac{1}{\log n}\right) = n(H_{[\log n]})$.

Therefore, $T(n) = \theta(n\log_2\log_2 n)$.

f. Since $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right) + n = T(\frac{7n}{8})$, and choose the longest tree as the level number, we have # level is $\log n$, the total cost is $T(n) = \sum_{i=1}^{\log n}(\frac{7}{8})^i n + cn \le \sum_{i=1}^{\infty}\left(\frac{7}{8}\right)^i n + cn = \frac{n}{1-\frac{7}{8}} + cn = 8n + cn$. So $T(n) = O(n)$

Check with lower bound: since $T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right) + n \ge n$, $T(n) = \Omega(n)$

Therefore, $T(n) = \theta(n)$

g. Since $T(n) = T(n-1) + \frac{1}{n}$, $T(1) = \frac{1}{1} = 1$, $T(2) = T(1) + \frac{1}{2} = 1 + \frac{1}{2}$, $T(3) = T(1) + T(2) + \frac{1}{3}$, we can see that: $T(n) = \sum_{i=1}^{n} \frac{1}{i} + \frac{1}{n}$. Since $\sum_{i=1}^{n} \frac{1}{i} = \ln n$, $T(n) = \theta(\log n)$

h. Since $T(n) = T(n-1) + \log n$, $T(1) = 0$, $T(2) = T(1) + \log 2$, $T(3) = T(1) + T(2) + \log 3$, $T(4) = T(1) + T(2) + T(3) + \log 4$, we have $T(n) = \sum_{i=1}^{n} \log i$.
Since $\sum_{i=1}^{n} \log i \leq n \log n$, we can guess $T(n) = \theta(n \log n)$.
Proof:
Upper bound: we have $T(n) = \sum_{i=1}^{n} \log i \leq \sum^{n} \log n = n \log n$, then $T(n) = O(n \log n)$.
Lower bound: we have $T(n) = \sum_{i=n/2}^{n} \log i \geq \frac{n}{2} \log \frac{n}{2} \geq \left(\frac{n}{2} - 1\right) \log n - \left(\frac{n}{2} - 1\right) = \Omega(n \log n)$.
Therefore, $T(n) = \theta(\log n)$.

i. Since $T(n) = T(n-2) + \frac{1}{\log n} = T(n-4) + \frac{1}{\log(n-2)} + \frac{1}{\log n} = \frac{1}{\log_2 n} + \frac{1}{\log_2(n-2)} + \frac{1}{\log_2(n-4)} + \cdots = \begin{cases} T(0) + \sum_{i=1}^{\frac{n}{2}} \frac{1}{\log_2(2i)} & n \text{ is even} \\ T(1) + \sum_{i=1}^{\frac{n}{2}+1} \frac{1}{\log_2(2i+1)} & n \text{ is odd} \end{cases}$. So $T(n) \geq \frac{n}{2} \times \frac{1}{\log_2(n)}$, then $T(n) = \Omega(n/\log n)$. For upper bound, $T(n) = \sum_{i=1}^{\frac{\sqrt{n}}{2}} \frac{1}{\log_2(2i)} + \sum_{i=\frac{\sqrt{n}}{2}+1}^{\frac{n-\sqrt{n}}{2}} \frac{1}{\log_2(2i)}$.
Since $\frac{1}{\log_2(2i)} \leq 1, \sum_{i=1}^{\frac{\sqrt{n}}{2}} \frac{1}{\log_2(2i)} \leq \frac{\sqrt{n}}{2}$. Since $\frac{1}{\log_2(2i)} \leq \frac{2}{\log n}$ $where\ i \geq \frac{\sqrt{n}}{2}$, $\sum_{i=\frac{\sqrt{n}}{2}}^{\frac{n-\sqrt{2}}{2}} \frac{1}{\log_2(2i)} \leq \frac{n-\sqrt{n}}{2} \times \frac{2}{\log n} = \frac{n-\sqrt{n}}{\log n}$. Therefore, $T(n) \leq \frac{\sqrt{n}}{2} + \frac{n-\sqrt{n}}{\log n} = \frac{\sqrt{n}(\log n - 2) + n}{2 \log n}$.
So $T(n) = O(n/\log n)$, if $n \geq 4$. Therefore, $T(n) = \theta(n/\log n)$.

j. From $T(n)$ recursion tree, we can see the root cost is n; 1st level, the cost = $\sqrt{n} \times \sqrt{n} = n$; 2nd level, the cost = $n^{\frac{1}{2}} \times n^{\frac{1}{4}} = n^{\frac{3}{4}}$. The ith level, the cost = $n^{1-\frac{1}{2^i}}$. Therefore, the cost of any level would not be greater than n. Learning from problem 4-3 (d), we know the number of level of the recursive tree is $\log \log n$. Therefore, we can guess that $T(n) = \theta(n \ \log \log n)$.
Proof:
We substitute $n \ \log \log n$ to $T(n)$. We have: $T(n) = \sqrt{n}\left(\sqrt{n} \ \log \log \sqrt{n}\right) + n = n \ \log \log \sqrt{n} + n = n(-1 + \ \log \log n) + n = n \ \log \log n$. Therefore $T(n) = \theta(n \ \log \log n)$.