Recursion Tree Method

Ibrahim Albluwi

General Idea. Guess a solution for the recurrence as follows.

1. Draw a recursion tree to visualize the amount of work done.
2. Find the number of levels in the tree.
3. Find the amount of work done at each level.
4. Sum the work done at all levels.

A General Example. Consider the following (special case) recurrence equation.

$$T(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ aT(\frac{n}{b}) + f(n) & \text{if } n > 1 \end{cases}$$

The recursion tree is a complete tree with the following properties.

1. Number of levels $=$ height $+ 1$.
2. Height $= \log_b n$
3. Number of nodes at level $i = a^i$
4. Work done at level $i = a^i \bullet f(\frac{n}{b^i})$
5. Total amount of work $= \sum_{i=0}^{\text{height}} a^i \bullet f(\frac{n}{b^i})$

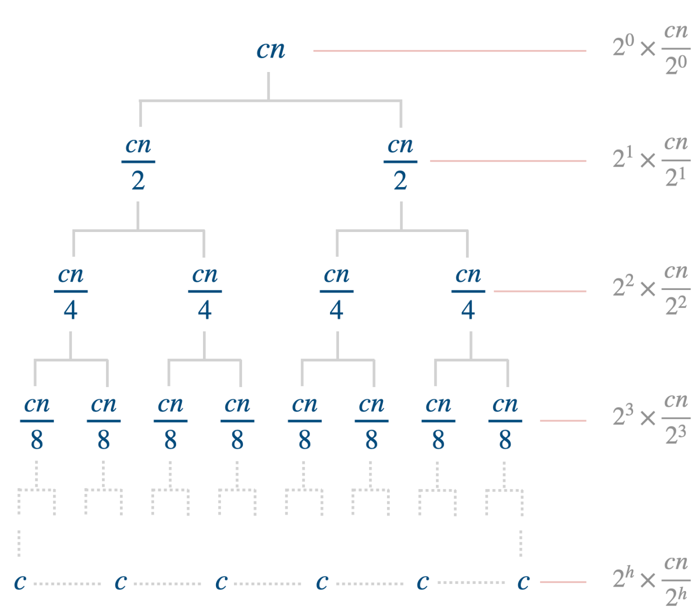
A Familiar Example. Consider the following simplified recurrence equation for Merge Sort.

$$T(n) = \begin{cases} c & \text{if } n \leq 1 \\ 2T(\frac{n}{2}) + cn & \text{if } n > 1 \end{cases}$$

1. Number of levels $=$ height $+ 1$.
2. Height $= \log_2 n$
3. Number of nodes at level $i = 2^i$
4. Work done at level $i = 2^i \times \frac{cn}{2^i} = cn$
5. Total amount of work $= \sum_{i=0}^{\log_2 n} cn$

$= cn \times (\log_2 n + 1)$

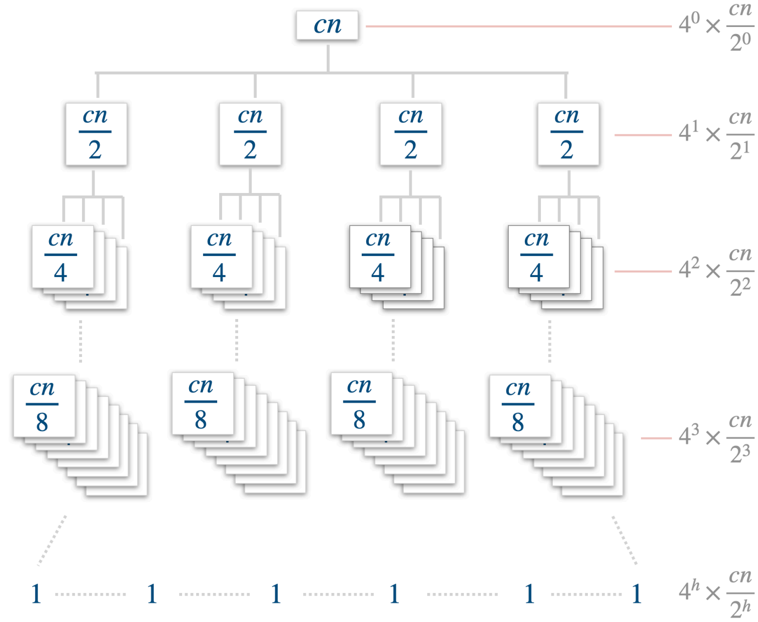
$= cn \log_2 n \; + \; cn = \Theta(n \log n)$



Example 1. $T(n) = 4T(\frac{n}{2}) + cn$ if $n > 1$, c if $n \leq 1$

- Height $= \log_2 n$
- Number of nodes at level $i = 4^i$
- Work done at level $i = 4^i \times \dfrac{cn}{2^i} = 2^i \times cn$
- Total amount of work:

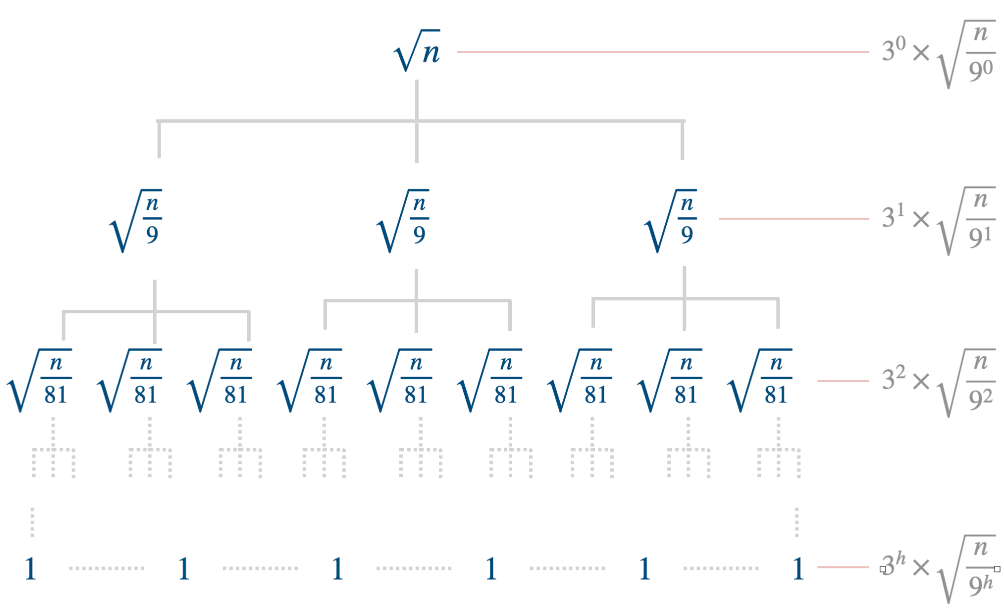
$$= \sum_{i=0}^{\log_2 n} (2^i \times cn) = cn \times \sum_{i=0}^{\log_2 n} 2^i$$
$$= cn \times (2^{\log_2 n+1} - 1)$$
$$= cn \times 2^{\log_2 n+1} - cn$$
$$= cn \times 2 \times 2^{\log_2 n} - cn$$
$$= cn \times 2 \times n - cn$$
$$= \Theta(n^2)$$



Example 2. $T(n) = 3T(\frac{n}{9}) + \sqrt{n}$ if $n > 1$, 1 if $n \leq 1$

- Height $= \log_9 n$
- Number of nodes at level $i = 3^i$
- Work done at level $i = 3^i \times \sqrt{\dfrac{n}{9^i}} = 3^i \times \sqrt{\dfrac{n}{(3^i)^2}} = 3^i \times \dfrac{\sqrt{n}}{3^i} = \sqrt{n}$

Total amount of work $= \displaystyle\sum_{i=0}^{\log_9 n} \sqrt{n} = \sqrt{n} \times (\log_9 n + 1) = \sqrt{n}\log_9 n + \sqrt{n} = \Theta(\sqrt{n}\log n)$



Example 3. $T(n) = 2T(\frac{n}{2}) + n^2$ if $n > 1$, 1 if $n \le 1$

- Height $= \log_2 n$
- Number of nodes at level $i = 2^i$
- Work at level $i =$

$$2^i \times (\frac{n}{2^i})^2 = \frac{n^2}{2^i} = (\frac{1}{2})^i \times n^2$$

- Total Work $=$

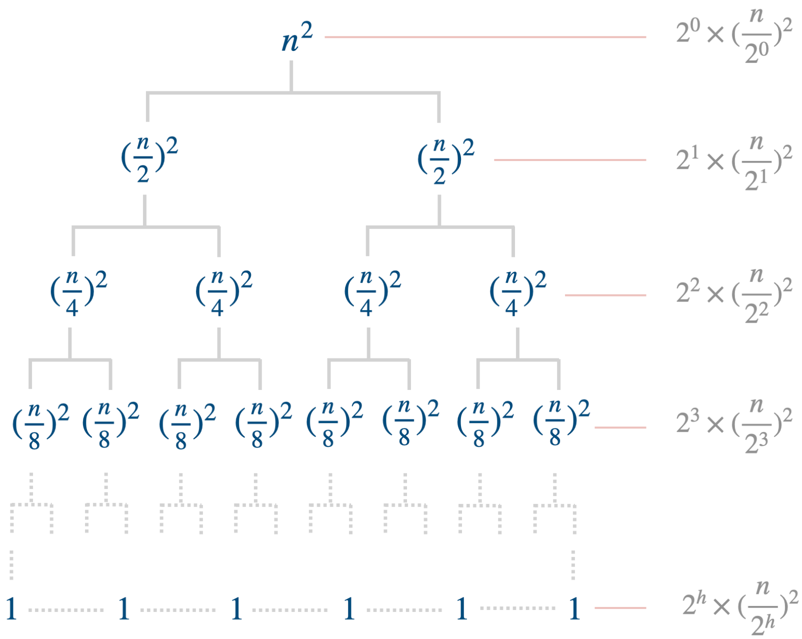
$$= \sum_{i=0}^{\log_2 n} (\frac{1}{2})^i \times n^2 = n^2 \times \sum_{i=0}^{\log_2 n} (\frac{1}{2})^i$$

$$= n^2 \times \frac{(\frac{1}{2})^{\log_2 n+1} - 1}{\frac{1}{2} - 1}$$

$$= n^2 \times \frac{\frac{1}{2} \times (\frac{1}{2})^{\log_2 n} - 1}{-\frac{1}{2}}$$

$$= n^2 \times (2 - (\frac{1}{2})^{\log_2 n})$$

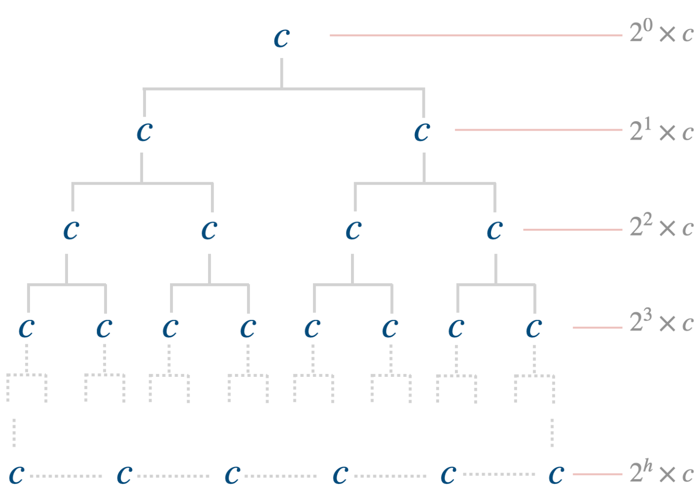
$$= n^2 \times (2 - \frac{1}{n}) = \Theta(n^2)$$



Example 4. $T(n) = 2T(\frac{n}{2}) + c$ if $n > 1$, c if $n \le 1$

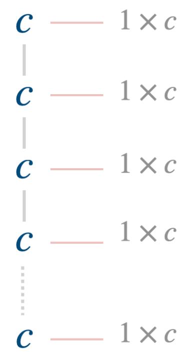
- Height $= \log_2 n$
- Number of nodes at level $i = 2^i$
- Work done at level $i = 2^i \times c$

Total amount of work $= \sum_{i=0}^{\log_2 n} 2^i \times c = c \times \sum_{i=0}^{\log_2 n} 2^i = c \times (2^{\log_2 n+1} - 1) = \Theta(n)$



Example 5. $T(n) \;=\; T(\frac{n}{2}) + c \quad \text{if } n > 1, \quad c \quad \text{if } n \le 1$

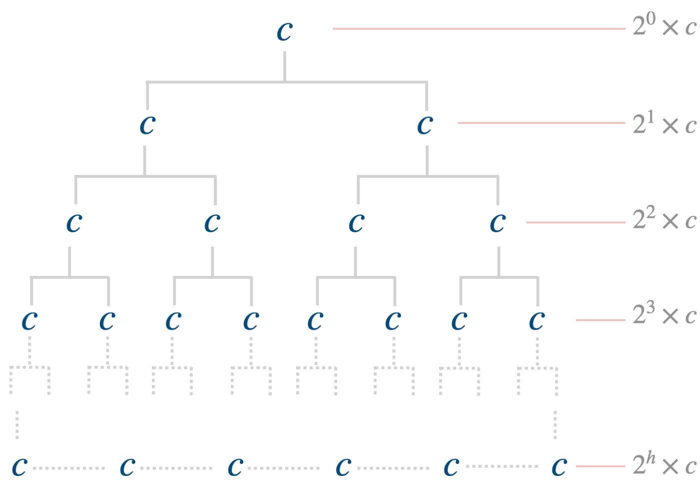
- Height $= \log_2 n$
- Number of nodes at level $i = 1^i = 1$
- Work done at level $i = 1 \times c$



$$\text{Total amount of work} = \sum_{i=0}^{\log_2 n} c \;=\; c \times (\log_2 n + 1)$$
$$= c \log_2 n + c = \Theta(\log n)$$

Example 6. $T(n) \;=\; 2T(n-1) + c \quad \text{if } n > 1, \quad c \quad \text{if } n \le 1$

- Height $= n - 1$
- Number of nodes at level $i = 2^i$
- Work done at level $i = 2^i \times c$



$$\text{Total amount of work} = \sum_{i=0}^{n-1} c \times 2^i$$
$$= c \times (2^n - 1)$$
$$= \Theta(2^n)$$

Example 7. $T(n) \;=\; 3T(\frac{n}{4}) + n^2 \quad \text{if } n > 1, \quad 1 \quad \text{if } n \le 1$

- Height $= \log_4 n$
- Number of nodes at level $i = 3^i$
- Work done at level $i = 3^i \times (\frac{n}{4^i})^2 \;=\; (\frac{3}{16})^i \times n^2$

$$\text{Total amount of work} = \sum_{i=0}^{\log_4 n} (\frac{3}{16})^i \times n^2 \;=\; n^2 \times \sum_{i=0}^{\log_4 n} (\frac{3}{16})^i \;\le\; n^2 \times \sum_{i=0}^{\infty} (\frac{3}{16})^i$$
$$\le\; n^2 \times (\frac{1}{1 - \frac{3}{16}}) \le \frac{n^2}{\frac{13}{16}} \;\le\; \frac{16}{13} n^2 \;=\; O(n^2)$$

The total work is also clearly $\Omega(n^2)$ since the work at level $0 = n^2$. Hence, the total work is $\Theta(n^2)$

Note. Consider the sum $ak^0 + ak^1 + ak^2 + ak^3 + \ldots + ak^n = \sum_{i=0}^{n} ak^i$.

If $-1 < k < 1$, then the sum is less than $\sum_{i=0}^{\infty} ak^i = \frac{a}{1-k}$

Example 1. $\sum_{i=0}^{n} (\frac{1}{2})^i \leq \sum_{i=0}^{\infty} (\frac{1}{2})^i = \frac{1}{1-\frac{1}{2}} = 2$

Example 2. $\sum_{i=0}^{n} 5(\frac{2}{3})^i \leq \sum_{i=0}^{\infty} 5(\frac{2}{3})^i = \frac{5}{1-\frac{2}{3}} = 15$

Exercises. Solve the following recurrence equations using the recursion tree method.

1. $T(n) = 2T(\frac{n}{4}) + cn$ if $n > 1$, c if $n \leq 1$.

2. $T(n) = 3T(\frac{n}{2}) + cn$ if $n > 1$, c if $n \leq 1$.

3. $T(n) = T(\frac{n}{2}) + n^2$ if $n > 1$, 1 if $n \leq 1$.

4. $T(n) = T(\frac{n}{2}) + \log_2 n$ if $n > 1$, 0 if $n \leq 1$.

5. $T(n) = 2T(\frac{n}{2}) + n\log_2 n$ if $n > 1$, 0 if $n \leq 1$.

6. $T(n) = 4T(\frac{n}{2}) + n^2$ if $n > 1$, 1 if $n \leq 1$.

7. $T(n) = 2T(n-1) + n$ if $n > 1$, 1 if $n \leq 1$. (note: $\sum_{i=0}^{n} i \times 2^i = (n-1)2^{n+1} + 2$)

8. $T(n) = 2T(\frac{n}{2}) + \log_2 n$ if $n > 1$, 0 if $n \leq 1$.