Recurrence Relations

Recurrence relation

$$\begin{aligned} T(n) &= T(n-1) + f(n) \\ &= T(n-2) + f(n-1) + f(n) \\ &= \cdots \\ &= T(0) + \sum_{j=1}^{n} f(j). \end{aligned}$$

For a specific function $f(x)$, the sum $\sum_{j=1}^{n} f(j)$ can usually be either computed exactly or its order of growth ascertained. For example, if $f(n) = 1$, $\sum_{j=1}^{n} f(j) = n$; if $f(n) = \log n$, $\sum_{j=1}^{n} f(j) \in \Theta(n \log n)$; if $f(n) = n^k$, $\sum_{j=1}^{n} f(j) \in \Theta(n^{k+1})$. The sum $\sum_{j=1}^{n} f(j)$ can also be approximated by formulas involving integrals (see, in particular, the appropriate formulas in Appendix A).

Asymptotic property

Symmetry:

$$f(n) = \Theta(g(n)) \text{ if and only if } g(n) = \Theta(f(n)) .$$

Transpose symmetry:

$$f(n) = O(g(n)) \text{ if and only if } g(n) = \Omega(f(n)) ,$$
$$f(n) = o(g(n)) \text{ if and only if } g(n) = \omega(f(n)) .$$

Transitivity:

$$
\begin{array}{llll}
f(n) = \Theta(g(n)) & \text{and } g(n) = \Theta(h(n)) & \text{imply} & f(n) = \Theta(h(n))\,, \\
f(n) = O(g(n)) & \text{and } g(n) = O(h(n)) & \text{imply} & f(n) = O(h(n))\,, \\
f(n) = \Omega(g(n)) & \text{and } g(n) = \Omega(h(n)) & \text{imply} & f(n) = \Omega(h(n))\,, \\
f(n) = o(g(n)) & \text{and } g(n) = o(h(n)) & \text{imply} & f(n) = o(h(n))\,, \\
f(n) = \omega(g(n)) & \text{and } g(n) = \omega(h(n)) & \text{imply} & f(n) = \omega(h(n))\,.
\end{array}
$$

Reflexivity:

$$
\begin{aligned}
f(n) &= \Theta(f(n))\,, \\
f(n) &= O(f(n))\,, \\
f(n) &= \Omega(f(n))\,.
\end{aligned}
$$

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

Substituting $n = 2^m$, we get:

$$T(n) = T(2^m) = T(\sqrt{2^m}) + \log 2^m = T(2^{m/2}) + m$$

Let $S(m)$ be defined as $S(m) = T(2^m) = T(2^{m/2}) + m$. (*)

Note that $S(m/2) = T(2^{m/2})$, so by (*):

$$S(m) = S(m/2) + m.$$

You can now use the Master Theorem to solve this recursion in terms of $S(m)$, to obtain:

$$S(m) = \Theta(m)$$

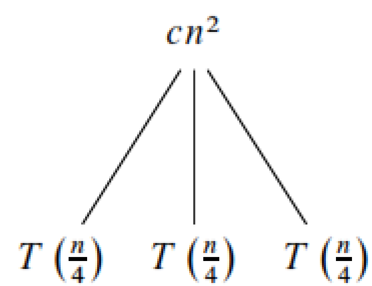
You could have deduced this result also by observing that in your recursion, combining the sub-problems takes $\Theta(m)$ time, which dominates the time it takes for the recursion to run (logarithmic time).

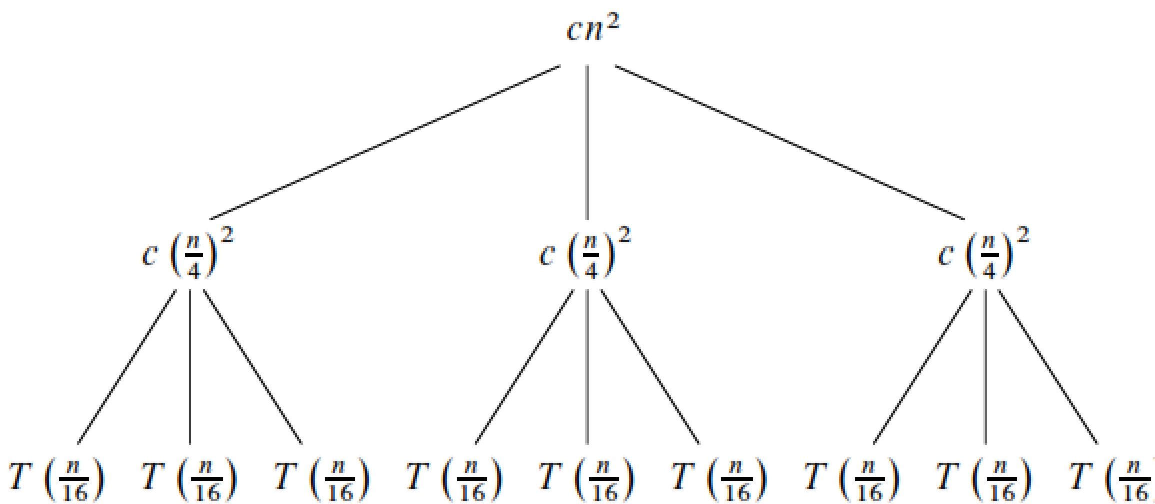
To finish, we back-substitute $m = \log n$ to obtain:

$$S(m) = T(2^m) = T(2^{\log n}) = T(n) = \Theta(m) = \Theta(\log n).$$

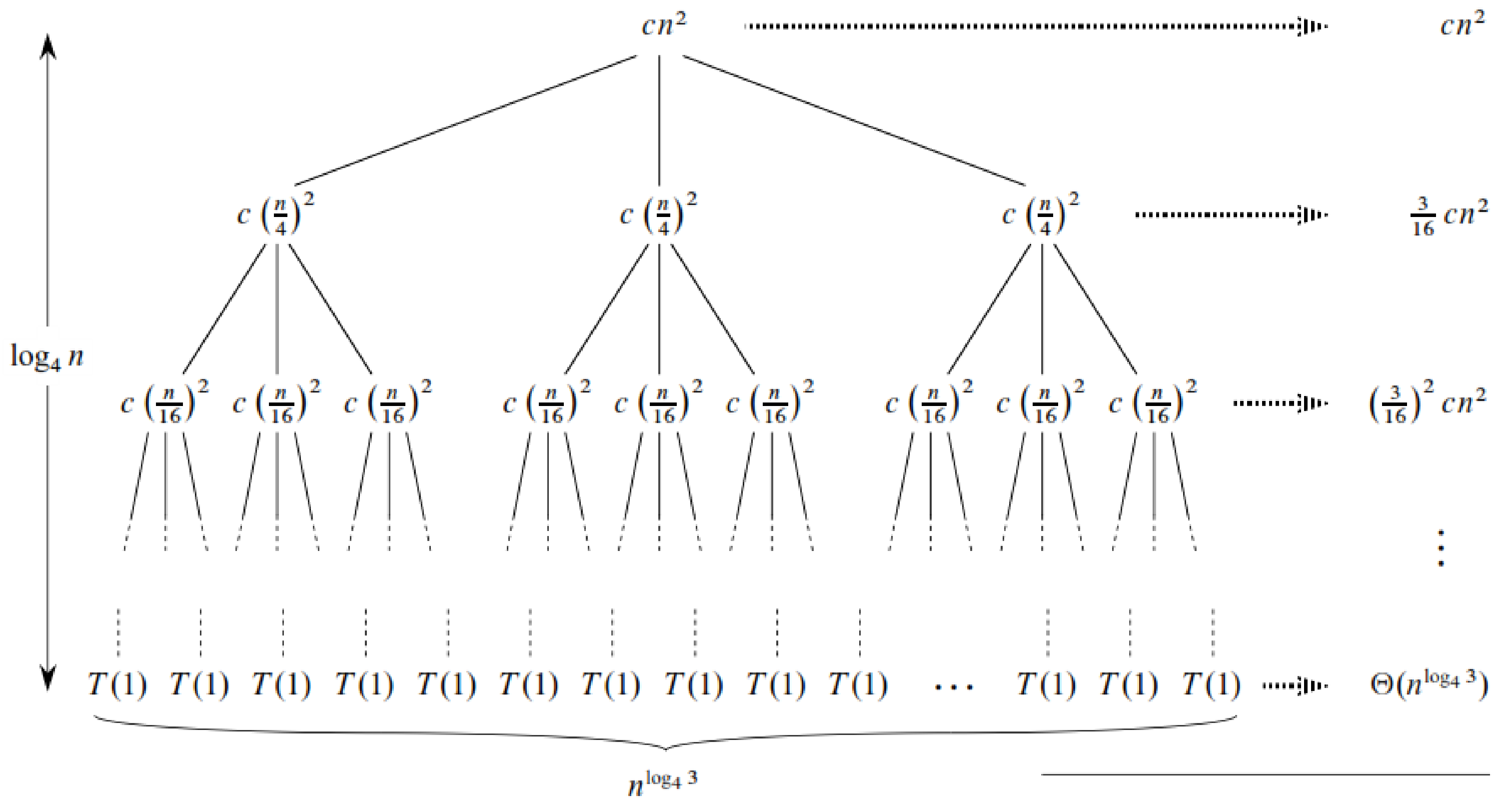
Clearing away the clutter, $T(n) = \Theta(\log n)$

$$T(n) = 3T(n/4) + cn^2.$$


cn^2
$T\left(\frac{n}{4}\right)$ $T\left(\frac{n}{4}\right)$ $T\left(\frac{n}{4}\right)$



cn^2
$c\left(\frac{n}{4}\right)^2$ $c\left(\frac{n}{4}\right)^2$ $c\left(\frac{n}{4}\right)^2$
$T\left(\frac{n}{16}\right)$ $T\left(\frac{n}{16}\right)$ $T\left(\frac{n}{16}\right)$ $T\left(\frac{n}{16}\right)$ $T\left(\frac{n}{16}\right)$ $T\left(\frac{n}{16}\right)$ $T\left(\frac{n}{16}\right)$ $T\left(\frac{n}{16}\right)$ $T\left(\frac{n}{16}\right)$


cn^2
$c\left(\frac{n}{4}\right)^2$
$c\left(\frac{n}{16}\right)^2$
$\log_4 n$
$T(1)$
$n^{\log_4 3}$
$\frac{3}{16}cn^2$
$\left(\frac{3}{16}\right)^2 cn^2$
$\Theta(n^{\log_4 3})$