It is sometimes possible to do these calculations with a little more rigour and obtain an asymptotic solution directly. We call this the iteration method.

Using the same example:

\[ T(n) = \begin{cases} 
1 & 1 \leq n < 3 \\
2^{T\left(\left\lfloor \frac{n}{3} \right\rfloor \right)} + n & n \geq 3 
\end{cases} \]

We no longer assume that \( n \) is an exact power of 3. We substitute this expression into itself:

\[ T(n) = n + 2^{T\left(\left\lfloor \frac{n}{3} \right\rfloor \right)} \]

\[ = n + 2^{\left(\left\lfloor \frac{n}{3} \right\rfloor \right) + 2^{T\left(\left\lfloor \frac{\left\lfloor n/3 \right\rfloor}{3} \right\rfloor \right)}} \]

\[ = n + 2^{\left\lfloor n/3 \right\rfloor} + 2^{2^{\left\lfloor n/3 \right\rfloor}} + 2^{3^{T\left(\left\lfloor \left\lfloor n/3 \right\rfloor /3 \right\rfloor \right)}} \]

\[ = n + 2^{\left\lfloor n/3 \right\rfloor} + 2^{2^{\left\lfloor n/3 \right\rfloor}} + 2^{3^{T\left(\left\lfloor n/3 \right\rfloor /3 \right\rfloor}} \]

\[ \vdots \]

\[ = \sum_{i=0}^{k-1} 2^{\left\lfloor \frac{n}{3^i} \right\rfloor} + 2^k T\left(\left\lfloor \frac{n}{3^k} \right\rfloor \right) \]
WE CHOOSE $k$ SO THAT:

$$1 \leq \left\lfloor \frac{n}{3^k} \right\rfloor < 3$$

$$\therefore \quad 1 \leq \frac{n}{3} \leq 3$$

$$\therefore \quad 3^k \leq n < 3^{k+1}$$

$$\therefore \quad k \leq \log_3 n < k+1$$

$$\therefore \quad k = \left\lfloor \log_3 (n) \right\rfloor .$$

WITH THIS VALUE OF $k$ WE HAVE

$$T(n) = \sum_{i=0}^{k-1} 2^i \left\lfloor \frac{n}{3^i} \right\rfloor + 2^k \\ 
\leq n \left( \sum_{i=0}^{k-1} \left( \frac{2}{3} \right)^i \right) + 2^k \log_3 n \\ 
= n \left( \frac{1 - \left( \frac{2}{3} \right)^k}{1 - \left( \frac{2}{3} \right)} \right) + n \log_3 2 \\ 
= 3n \left( 1 - \left( \frac{2}{3} \right)^k \right) + n \log_3 2 \\ 
\leq 3n + n \log_3 n = O(n)$$

$\therefore \quad T(n) = O(n)$ HAS BEEN PROVED. ALSO

$$T(n) = \sum_{i=0}^{k-1} 2^i \left\lfloor \frac{n}{3^i} \right\rfloor + 2^k \\ 
\geq \sum_{i=0}^{k-1} 2^i \left( \frac{n}{3^i} - 1 \right) + 2^k$$
\[ \begin{align*}
&= n \left( \sum_{i=0}^{n} \left( \frac{1}{2} \right)^i \right) - \sum_{i=0}^{k-1} 2^i + 2^k \\
&\geq n - \left( \frac{2^k - 1}{2-1} \right) + 2^k \\
&= n + 1 - 2^k + 2^k \\
&= n + 1 = \Omega(n). \\
\end{align*} \]

\[ \therefore T(n) = \Omega(n), \text{ Whence } T(n) = \Theta(n). \]

The iteration method can sometimes be used to find exact solutions to recurrence relations (as opposed to asymptotic solutions.)

\[ \begin{align*}
\text{Ex.} \quad T(n) &= \begin{cases} 
3 & 0 \leq n < 2 \\
T(n-2) + n & n \geq 2 
\end{cases} \\
T(n) &= n + T(n-2) \\
&= n + (n-2) + T(n-4) \\
&= n + (n-2) + (n-4) + T(n-6) \\
&\vdots \\
&= \sum_{i=0}^{k-1} (n-2i) + T(n-2k) 
\end{align*} \]
Choose \( k \) so that

\[
0 \leq n - 2k < 2
\]

\[
\therefore 2k \leq n < 2k + 2
\]

\[
\therefore k \leq \frac{n}{2} < k + 1
\]

\[
\therefore k = \left\lfloor \frac{n}{2} \right\rfloor
\]

Then \( T(n-2k) = 3 \), \( \forall n \) such that

\[
T(n) = n \left( \sum_{i=0}^{\frac{k-1}{2}} 1 \right) - 2 \sqrt{\sum_{i=0}^{\frac{k-1}{2}} i} + 3
\]

\[
= kn - 2 \left( \frac{k(k-1)}{2} \right) + 3
\]

\[
= kn - k(k-1) + 3
\]

Hence

\[
T(n) = \left\lfloor \frac{n}{2} \right\rfloor n - \left\lfloor \frac{n}{2} \right\rfloor \left( \left\lfloor \frac{n}{2} \right\rfloor - 1 \right) + 3
\]

Exercise

1) \( \lfloor f(n) \rfloor = \Theta (f(n)) \)

2) \( \Theta (f(n)), \Theta (g(n)) = \Theta (f(n) \cdot g(n)) \).

It follows that \( T(n) = \Theta (n^2) \). One can also show by some algebra that

\[
T(n) = 3 + \frac{n(n+2)}{4} + \frac{3}{8} (n^2 - 1)
\]
4.3 The Master Theorem

This is a method for finding (asymptotic) solutions to recurrences of the form

\[ T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n) \]

where \( a \geq 1 \), \( b > 1 \) and \( f(n) \) is asymptotically positive. Here, \( T\left(\frac{n}{b}\right) \) denotes either \( T\left(\frac{n}{b^2}\right) \) or \( T\left(\frac{n}{b^2}\right) \), and it is understood that \( T(n) = \Theta(1) \) for some finite set of initial terms.

Such a recurrence describes the running time of a "divide and conquer" algorithm which divides a problem of size \( n \) into \( a \) subproblems, each of size \( \frac{n}{b} \). Here, \( f(n) \) represents the cost of doing the dividing and recombining.

**Master Theorem**

Let \( a \geq 1 \), \( b > 1 \), \( f(n) \) asymptotically positive, and let \( T(n) \) be defined by (*). Then

1. If \( f(n) = \Theta\left(n^{\log_b a - \epsilon}\right) \) for some \( \epsilon > 0 \)
   
   Then
   
   \[ T(n) = \Theta\left(n^{\log_b a}\right) \]
(2) If \( f(n) = \Theta(n^{\log_\alpha a}) \), then
\[
\frac{1}{f(n)} = \Theta\left(n^{\log_\alpha a} \cdot \log(n)\right).
\]

(3) If \( f(n) = \Omega\left(n^{\log_\alpha a + \varepsilon}\right) \) for some \( \varepsilon > 0 \), and if \( a f(n) \leq c f(n) \) for some \( c < 1 \) and sufficiently large \( n \), then
\[
\frac{1}{f(n)} = \Theta(f(n)).
\]

**Remarks**

In each case we compare \( f(n) \) to the function \( n^{\log_\alpha a} \), and the solution is determined by which is larger.

In Case (1) \( n^{\log_\alpha a} \) is polynomially larger and the solution is \( \Theta(n^{\log_\alpha a}) \). In Case (3) \( f(n) \) is larger (and an additional regularity condition is met) and the solution is \( \Theta(f(n)) \). In Case (2) the two functions are asymptotically equivalent and the solution is \( \Theta(f(n) \cdot \log n) \).

To say that \( f(n) \) is polynomially smaller than \( n^{\log_\alpha a} \) (as in (1)) means that the two functions differ by a factor \( n^\varepsilon \) for some \( \varepsilon > 0 \).
Note that the three cases do not cover all possibilities, there is a gap between cases (1) and (2) when \( t(n) \) is smaller than \( n^{\log_b a} \) but not polynomially smaller.

\[
T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n}
\]

\[
\log_2 2 = 1, \quad \frac{n}{\log n} \neq \mathcal{O}(n^{1-\varepsilon}) \text{ for any } \varepsilon > 0.
\]

Similar comments hold for cases (2) and (3). In addition, the regularity condition in (3) may fail to hold.

**Example**

\[
T(n) = 8T\left(\frac{n}{2}\right) + n^3
\]

\[
a = 8, \quad b = 2, \quad \log_b a = 3, \quad t(n) = n^3 = \mathcal{O}(n^{\log_b a})
\]

\[\therefore \text{ case (2)}.
\]

\[\therefore T(n) = \mathcal{O}(n^3 \log(n)).\]