16.5 Time Complexity using Masters Theorem

Time Complexity

Recurrence Relation

A recurrence relation is an equation that recursively defines a sequence.

Let's see it with an example

Fibonacci Series:

$$F(n) = F(n-1) + F(n-2)$$

Master Theorem

Gives the Time Complexity for the recurrence relation:

$$T(n) = aT(n/b) + f(n)$$

Master Theorem

For the Recurrence: $T(n) = aT(n/b) + \Theta(n^c)$, $a >= 1$, $b > 1$

There are following three cases:

1. If $f(n) = \Theta(n^c)$ where $c < \log_b a$ then $T(n) = \Theta(n^{\log_b a})$
2. If $f(n) = \Theta(n^c)$ where $c = \log_b a$ then $T(n) = \Theta(n^c \log n)$
3. If $f(n) = \Theta(n^c)$ where $c > \log_b a$ then $T(n) = \Theta(f(n))$

Problems:

1. $T(n) = 2T(n/2) + \Theta(n)$

   $a=2$, $b=2$, $c=1$

   $\rightarrow c = \log_b a$

   Time Complexity: $\Theta(n \log_2 n)$

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

   $a=3$, $b=2$, $c=2$

   $\rightarrow c > \log_b a$

   Time Complexity: $\Theta(n^2)$

Recurrence Tree Method:

1. $T(n) = T(n-1) + n$

$$T(n) = T(n-1) + n$$
$$T(n-1) = T(n-2) + n-1$$
$$T(n-2) = T(n-3) + n-2$$
$$\vdots$$
$$T(1) = 1$$

Adding all the terms, we get

$$T(n) = n + (n-1) + (n-2) + (n-3) + \dots + 1$$

$$T(n) = (n * (n+1))/2$$

$$T(n) = \Theta(n^2)$$