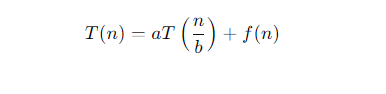
**Recurrence Relations - A Complete Guide**

**General Form of a Recurrence Relation:**



**Where:**

* **T(n)** is the **time complexity of** the **problem** of **size n**
* **a is the number of subproblems**
* **b is the factor** **by which the problem size is reduced in each subproblem**
* **f(n)** is the cost of **dividing the problem and** **combining the subproblems** (the work done outside the recursive calls).

**Methods for Solving Recurrence Relations**

a) **Substitution Method**

b) **Recursion Tree Method :** The recursion tree method visualizes the recurrence as a tree, where each node represents a subproblem.

**c) Master Theorem**

The **Master Theorem** provides a direct way to solve recurrences of the form:



**T(n) = aT(n/b) + f(n) where a >= 1 and b > 1**