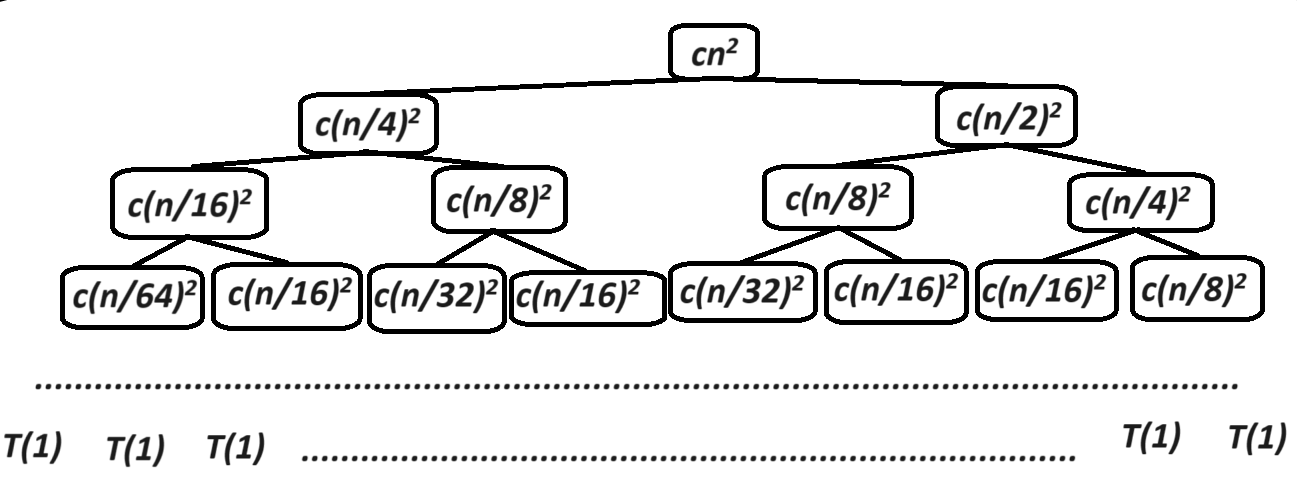
**Chapter 3: ALGORITHMIC METHODS**

**Topic – 1: Recurrence Method**

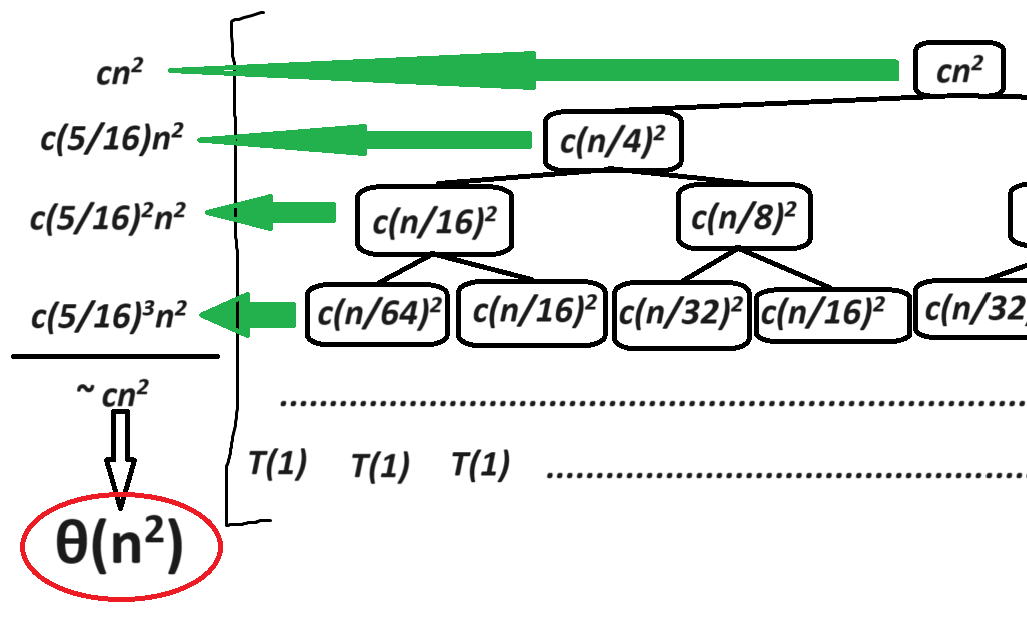
**Our Example**

**T(n) = T(n/4) + T(n/2) + cn2**

**Example’s Diagram**



**Right Side Of Tree**



* Notice that the **left side** of the tree **ends first** for producing smaller values down the level, so left side has **shorter height**.

**n = 4h1**

**h1 = log4(n) = log2(n).logn(2)**

**h2 = log2(n)**

**Final Equation**

**T(n) = T(n/4) + T(n/2) + cn2 = θ(n2)**

**Topic – 2: Substitution Method**

**Steps Involved**

* **Step 1:** Guess the solution.
* **Step 2:** Verify the guess by the method of **mathematical induction**.
* **Step 3:** Make the **bound tighter** by guessing even smaller solution.

**Mathematical Induction**

**Suppose a function P(n), where n Є N = 1, 2, 3, …, infinity**

**Base case: P(1) is true.**

**k = 1, 2, 3, …, n-1**

**Induction hypothesis:**

**Assuming P(k) is true for k < n.**

**So, P(n) is also true for n.**

**Example**

**Given: T(n) = 4T(n/2) + n**

**Guess: T(n) = O(n3)**

**Assumption:**

**T(k) = O(k3) [for all k<n (k = 1, 2, …, n-1)]**

**T(k) ≤ ck3**

**To prove:**

**T(n) ≤ cn3**

**T(n) = O(n3)**

**Proof:**

**As per our assumption,**

**4T(n/2) + n ≤ 4c(n/2)3 + n**

**4c(n/2)3 + n**

**= cn3 + n**

**= cn3 – ((cn3/n) – n) ≥ 0**

**We must prove the second term as greater than 0, to prove the hypothesis.**

**For c > 2, [n is a natural number]**

**(cn3/n) – n ≥ 0**

**Thus, cn3 ≥ 0**

**By guessing even smaller value, we can make the bound tighter.**

**Like for: T(n) ≤ cn2 or T(n) ≤ c1n2 c2n**

**Note!**

**🡪 T(n) ≤ cn3 when T(n) = O(n3)**

**🡪 T(n) ≥ cn3 when T(n) = Ω(n3)**

**🡪 T(n) = cn3 when T(n) = θ(n3)**

**🡪 If you prove a hypothesis for both O and Ω of same equation, then the equation fits neatly into θ.**

**🡪 Recurrence tree is for guessing purpose, while substitution method is for proving the solution through induction.**