**Master Theorem**

The Master Theorem applies to recurrences of the following form:

**T (n) =**

Where a ≥ 1 and b > 1 are constants and f(n) is an asymptotically positive function.

These are 3 cases:

1. If f(n) = O (nlogb (a)-ϵ), then T(n) = Θ (nlogb a).

2. If f(n) = Θ (nlogb a), then T(n) = Θ (nlogb a \* log n).

3. If f(n) = Ω (nlogb (a)+ϵ), then T(n) = Θ (f(n)).

ϵ > 0 is a constant and f(n) satisfies the regularity condition, then T (n) = Θ(f(n)).

Regularity condition:  **≤** for some constant c < 1 and all sufficiently large n.

**Graded Coding Assignment 2 Part 1**

1. **T (n) =**

T (n) =

a = 3, b = 2, *f*(n) = *n*, d = 1

log b a = log2 3 ≈ 1.58 > d = 1

Now, 1 i.e., d can be written as

1.58 – 0.58 or log2 (3) – 0.58 => logb (a)-ϵ

Therefore, **Case 1** of Master Theorem applies here,

Thus, **T (n) = Θ (nlog2 3) or Θ (nlog 3)**

1. **T (n) =**

T (n) =

a = 64, b = 8, *f*(n) =

Since,***f*(n) is not positive**, i.e., it’s negative

**Therefore, this recurrence cannot be solved using Master’s Theorem**

1. **T (n) =**

T (n) =

a = 2n, b = 12, *f*(n) =

Since,**a is not a constant**

**Therefore, this recurrence cannot be solved using Master’s Theorem**

1. **T (n) =**

T (n) =

a = 3, b = 3, *f*(n) = = = c.=> 1

log b a = log3 3 = 1

So, nlogb a = n1 = *f*(n)

Therefore, **Case 2** of Master Theorem applies here,

Thus, **T (n) = Θ (nlog3 3 \* log n) i.e., Θ (n log n)**

1. **T (n) = 7**

T (n) =

a = 7, b = 3, *f*(n) = 2, d = 2

log b a = log3 7 ≈ 1.77 < d=2

Now, 2 i.e., d can be written as

1.77 + 0.23 or log3 (7) + 0.23 => logb (a)+ϵ

Therefore, **Case 3** of Master Theorem applies here,

Thus, **T (n) = Θ (*f*(n)) i.e., Θ (n2)**

To prove *f*(n) is regular

Regularity condition:  **≤**

**≤**

**≤**

**≤**

**≤**

**True when 0.77 ≤ c < 1, therefore, *f*(n) = 2 is regular**