**Recurrence Relation (Contd.)**

#### Master Method

The Master Method is used for solving the following types of recurrence

T (n) = a TDAA Master Method+ f (n) with a≥1 and b≥1 be constant & f(n) be a function and DAA Master Methodcan be interpreted as

Let T (n) is defined on non-negative integers by the recurrence.

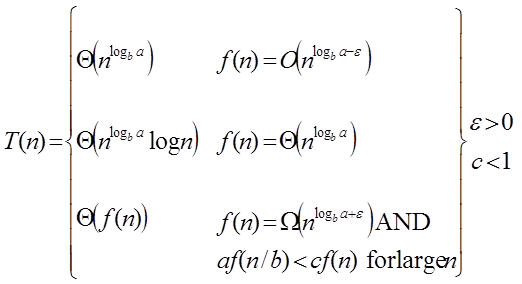
T (n) = a TDAA Master Method+ f (n)

In the function to the analysis of a recursive algorithm, the constants and function take on the following significance:

* n is the size of the problem.
* a is the number of subproblems in the recursion.
* n/b is the size of each subproblem. (Here it is assumed that all subproblems are essentially the same size.)
* f (n) is the sum of the work done outside the recursive calls, which includes the sum of dividing the problem and the sum of combining the solutions to the subproblems.
* It is not possible always bound the function according to the requirement, so we make three cases which will tell us what kind of bound we can apply on the function.

#### Master Theorem:

It is possible to complete an asymptotic tight bound in these three cases:



**Case1:** If f (n) = DAA Master Method for some constant ε >0, then it follows that:

T (n) = Θ DAA Master Method

**Example:**

T (n) = 8 T DAA Master Methodapply master theorem on it.

**Solution:**

Compare T (n) = 8 T DAA Master Method with

T (n) = a T DAA Master Method

a = 8, b=2, f (n) = 1000 n2, logba = log28 = 3

Put all the values in: f (n) = DAA Master Method

1000 n2 = O (n3-ε )

If we choose ε=1, we get: 1000 n2 = O (n3-1) = O (n2)

Since this equation holds, the first case of the master theorem applies to the given recurrence relation, thus resulting in the conclusion:

T (n) = Θ DAA Master Method

Therefore: T (n) = Θ (n3)

**Case 2:** If it is true, for some constant k ≥ 0 that:

F (n) = Θ DAA Master Methodthen it follows that: T (n) = Θ DAA Master Method

**Example:**

T (n) = 2 DAA Master Method, solve the recurrence by using the master method.

As compare the given problem with T (n) = a TDAA Master Method a = 2, b=2, k=0, f (n) = 10n, logba = log22 =1

Put all the values in f (n) =Θ DAA Master Method, we will get

10n = Θ (n1) = Θ (n) which is true.

**Therefore:** T (n) = Θ DAA Master Method

= Θ (n log n)

**Case 3:** If it is true f(n) = Ω DAA Master Method for some constant ε >0 and it also true that: a f DAA Master Method for some constant c<1 for large value of n, then:

1. T (n) = Θ((f (n))

**Example:** Solve the recurrence relation:

T (n) = 2 DAA Master Method

**Solution:**

Compare the given problem with T (n) = a T DAA Master Method

a= 2, b =2, f (n) = n2, logba = log22 =1

Put all the values in f (n) = Ω DAA Master Method..... (Eq. 1)

If we insert all the value in (Eq.1), we will get

n2 = Ω(n1+ε) put ε =1, then the equality will hold.

n2 = Ω(n1+1) = Ω(n2)

Now we will also check the second condition:

2 DAA Master Method

If we will choose c =1/2, it is true:

DAA Master Method∀ n ≥1

So it follows: T (n) = Θ ((f (n))

T (n) = Θ(n2)

**RELEVANT READING MATERIAL AND REFERENCES:**

**Source Notes:**

1. <https://www.javatpoint.com/daa-recurrence-relation>

**Lecture Video:**

1. <https://youtu.be/CyknhZbfMqc>
2. <https://youtu.be/OynWkEj0S-s>
3. <https://youtu.be/kGcO-nAm9Vc>

**Online Notes:**

1. <http://vssut.ac.in/lecture_notes/lecture1428551222.pdf>

**Text Book Reading:**

1. Cormen, Leiserson, Rivest, Stein, “*Introduction to Algorithms*”, Prentice Hall of India, 3rd edition 2012. problem, Graph coloring.

**In addition: PPT can be also be given.**