**Recurrence equations and their solution: substitution method**

The recurrence equation is an equation that defines a sequence recursively. It is normally in the form

T(n)=T(n-1) +n form>0(Recurrence relation n) T(0) =0 (Initial condition)

The general solution to the cursive function specifies some formula.

## SolvingRecurrenceEquations

The recurrence relation can be solved by following methods

* Substitution method
* Master’smethod
* Tree Recursion

**There are types of substitution**

* ***Forward substitution***
* ***Backward substitution***

## *Forward Substitution method*

This method makes use of an initial condition in the initial term, and a value for the next term is generated. This process is continued until some formula is guessed. Thus in this kind of method, we use reference equations to generate a few terms.

for example

Consider a recurrence relations(n) =T(n-1) +nwith initial conditions (0) =0LetT(n)=T(n-1)+n

Ifn =1 then

T(1)=T(0) +1=0+1 =1 (1)

Ifn =2then

T(2)=T(1) +2=1+2 =3 (2)

Ifn =3 then

T(3) =T(2) +3=3 +3 =6 (3)

Byobservingthe aboveequation,wecansay thatit isthe sum ofnnaturalnumber

T(n)=𝑛(𝑛+1)=𝑛2/2+𝑛

𝑛 2

SowecanwrittenasT(n) =O(n2)

## *BackwardSubstitutionMethod*

Inthismethod,backwardvalues aresubstitutedrecursivelyinorderto derivesomeformula.

ForExample

Consider, arecurrencerelationT(n)=T(n-1) +nwith initial conditionT(0) =0 (1)

Solution:

In Eqa(1) , to calculate T(n) , we need to know the value of T(n-1)T(n-1)= T(n-1-1) +(n-1)=T(n-2)+(n-1)

NowEqu(1) becomes T(n)=T(n-2)+(n-1) +n (2)

T(n-2)=T(n-2-1) +(n-2)=T(n-3)+(n-2)

NowEqa(2) becomesT(n) =T(n-3)+(n-2)+(n-1)+n (3)

Inthekthterms

T(n)=T(n-k)+(n-k+1)+(n-k+2)+-----+n (4)

Ifk =nin equ(4)then

T(n)=T(0)+1+2+3+ +n

T(n)=0+1+2+3+-----+n by substitutinginitialvalue T(0)=0

T(n)=𝑛(𝑛+1)=𝑛2/2+𝑛

𝑛 2

SoT(n)intermsofbigohnotationasT(n) =O(n2)

### Example:2

T(n)=T(n-1) +1withinitialconditionwith T(0)=0.Findbigohnotation.Solution:

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

T(n-1)=T(n-2)+1

Noweq(1)becomes T(n)=(T(n-2)+1)+1=T(n-2)+2 (2)

T(n-2)=T(n-3) +1

Noweq(2)becomes T(n)=(T(n-3)+1)+2=T(n-3)+3 (3)

So

T(n)=T(n-k)+k (4)

If k = n theneqa(4) becomesT(n ) = T(0) + n = 0 + n = nT(n) =O(n)

### Example3:

T(n)=2*T*(*n/*2) +*n.*T(1)=1asinitialconditionSolution:

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

*T*(*n/*2) = 2𝑇(𝑛/4)+𝑛/2

NowEq(1) becomes

T(n) =2[2𝑇(𝑛/4)+ 𝑛/2]+n = 4*T*(*n/4*)+n+n =4*T*(*n/4*)+2n (2)

*T*(*n/4*)=2𝑇(𝑛/8)+𝑛/4

NowEq (2)becomes

T(n) =4[2𝑇(𝑛/8)+ 𝑛/4]+2n = 8T(n/8)+n+2n =8T(n/8)+3n (3)

Equ(3)canbewrittenasT(n)=23T(n/23)+3n

Ingeneral

T(n) =2kT(n/2k) +kn (4)

Assume2k=n

NowEqu(4)canbewrittenasT(n)=n.T(n/n)+logn.n

=n.T(1) + n.lognT(n)=n+n.logn

i.eT(n)= O(n.logn)

### Example4:

T(n)=T(n/3)+CandinitialconditionT(1)=1Solution :

T(n) =T(n/3) +C (1)

T(n/3)=T(n/9)+C

NowEqu(1)becomes

T(n)=[T(n/9)+C]+C =T(n/9)+2C (2)

T(n/9)=T(n/27) +C

Now Equ(2) becomesT(n)=[T(n/27)+C]+2CT(n) =T(n/27) +3C

InGeneral

T(n) = T(n/3k) + kCPut 3k=n then

T(n)=T(n/n)+log3n.C

= T(1) + log3n.CT(n) =C. log3n +1

**Recurrence equations and their solution: Tree Recursion method**

In this method, we built a recurrence tree in which each node represents the cost of a single subproblem in the form of recursive function invocations. Then we sum up the cost at each level to determine the overall cost. Thus the recursion tree helps us to make a good guess of time complexity. The pattern is typically an arithmetic or geometric series.

**For example** consider the recurrence relationT(n)=T(n/4) +T(n/2)+cn2

cn2

/ \

T(n/4) T(n/2)

If we further breakdown the expressions T(n/4) and T(n/2), we get the following recursion tree.

c n2

/ \

c(n2)/16 c(n2)/4

/ \ / \

T(n/16)T(n/8) T(n/8)T(n/4)

Breaking down further gives us the followingcn2

/ \

c(n2)/16 c(n2)/4

/ \ / \

c(n2)/256c(n2)/64c(n2)/64c(n2)/16

/\ / \ / \ /\

To know the value of T(n), we need to calculate the sum of tree nodes level by level. If we sum the above treelevel by level, we get thefollowing series

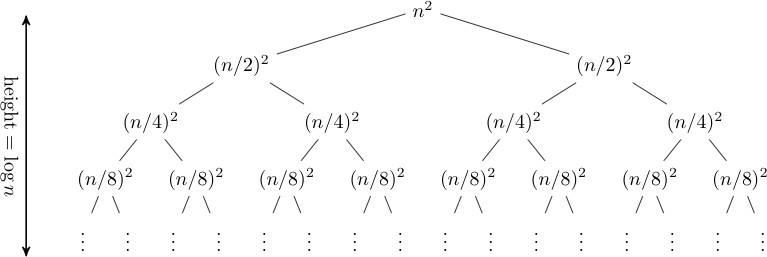
T(n)=c(n2+5(n2)/16 +25(n2)/256)+....

The above series is a geometrical progression with a ratio of 5/16. To get an upper bound, we can sum the infinite series. We get the sum as (n2)/(1 -5/16) which is O(n2)

### Example:

T(n) =2T(n/2) +n2.

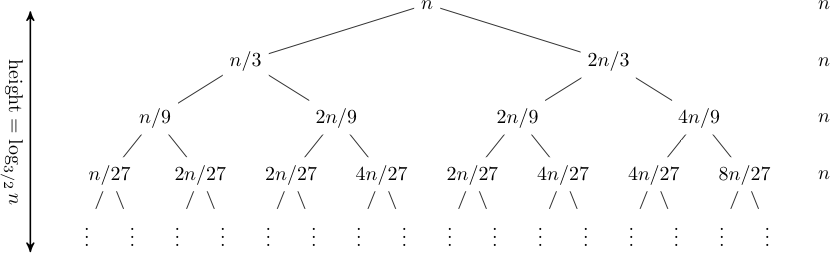
The recursion tree fort his recurrence has the following form:



Time complexity of above tree is O(n2)

**Let's consider another example**(n) =T(n/3) +T(2n/3) +n.

Expandingoutthefirstfewlevels,the recurrencetreeis:



Time complexity of above trees**(nlogn)**

**Recurrence equations and their solution:Master method**

We can solve the recurrence relation using a formula denoted by master’s method(n)=aT(n/b) + F(n) where n ≥ d and d is a constant

Then the master theorem can be stated for efficiency analysis as:

If F(n)is **ϴ**(nd) where d≥ 0

* Case1 : T(n)=**ϴ**(nd) if a< bd
* Case2: T(n)=**ϴ**(ndlogn)if a=bd
* Case3 :T(n)= **ϴ**(nlogba)if a> bd

**EXAMPLE.1:**T(n) =4T(n/2) +n

A=4,b =2,F(n) =n =n1i.e d =1

Compare a and bd,i.e 4 and21=4>2whichsatisfied case3:NowT(n)=**ϴ**(nlogba)=**ϴ**(nlog24)=**ϴ**(n2)

**Example 2:**T(n)=T(n/2)+1n2+n

2

A =1,b =2, d=2

Compareaandbd,i.e1and22=1<4whichsatisfiedcase1:T(n) =**ϴ**(nd)=**ϴ**(n2)

**Example 3 :**T(n)=2T(n/4) +√𝑛+42

A =2,b =4, d =½

Compareaand bd,i.e2and41/2=2=2 whichsatisfiedcase2:T(n) =**ϴ**(n1/2logn)=**ϴ**(√𝑛logn)

**Example 4 :** T(n) =3T(n/2) + 3n+1

4

A =3, b =2, d =1

Compareaandbd,i.e3and2=3 >2which satisfiedcase3:T(n) =**ϴ**(nlogba)==**ϴ**(nlog23)

## AnotherVariationofMaster’sMethod:

T(n)=aT(n/b)+ f(n)wheren ≥ d

* Case 1 : if f(n) is O(nlogba)andf(n) <nlogbathenT(n) =**ϴ**(nlogba)
* Case2 :if f(n)= **ϴ**(nlogbalogn)andf(n) =nlogbathen

T(n)=**ϴ**(nlogbalogn)

* Case3: iff(n) =Ω(nlogba) andf(n)>nlogbathenT(n)=**ϴ**(f(n))

Steps:

* 1. Getthevaluesof a,bandf(n)
  2. Determinethe value nloga

b

* 1. Comparef(n)and nloga

b

**Example:1**

T(n)=2T(n/2)+n

A =2, b =2, f(n)=n

Determine nlogba=nlog22=n1=n

Comparenlog22andf(n)i.en=nwhichiscase2:T(n) =**ϴ**(nlogbalogn)=**ϴ**(n1logn)=**ϴ**(nlogn)

**Example:2:**

T(n) =9T(n/3)+n

A =9, b =3,f(n)=n

Determine nlogba= nlog39= n2 andF(n)=n

Now f(n) <nlogbawhich is case 1:T(n) =**ϴ**(nlogba)=**ϴ**(nlog39)=**ϴ**(n2)

**Example:3:**

T(n)=3T(n/4)+nlognA=3,b=4,f(n)=nlogn

Determine nlogba =nlog43

f(n)>nlog43 whichiscase3:T(n) =**ϴ**(f(n)) =**ϴ**(nlogn)

**Example4:**

T(n) =3T(n/2)+n2

A =3, b =2,f(n)=n2

Determine nlogba= nlog23n2>nlog23case3:

T(n)=**ϴ**(f(n)) =**ϴ**(n2)

**Example5:**

T(n)= 4T(n/2) + n2A=4,b=2,f(n)=n2

Determine nlogba=nlog24=n2F(n)=n2case2:

T(n) = **ϴ**(nlogbalogn) =**ϴ**(nlog24logn)= **ϴ**(n2logn)**Example6:**

T(n) =4T(n/2)+n/logn

A = 4 , b = 2 ,f(n) = n/lognDetermine nlogba=nlog24=n2F(n)<n2case1 :

T(n) = **ϴ**(nlogba)= **ϴ**(nlog24)= **ϴ**(n2)**Example7 :**

T(n) =6T(n/3) +n2logn

A = 6 , b = 3 , f(n) = n2lognDetermine nlogba=nlog36=n2F(n)>nlogbacase3:

T(n) = **ϴ**(f(n)) =**ϴ**(n2logn)**Example8:(Need tobe solved)**

T(n)=4T(n/2)+cn case 1:T(n) =**ϴ**(n2)

**Example9:(Needtobe solved)**

T(n) =7T(n/3)+n2

T(n)=**ϴ**(n2) case3:

**Example10:(Needto besolved)**

T(n) =4T(n/2)+logn

T(n)=**ϴ**(nlogn) case