# Recurrence relation:

A recurrence relation, T(n), is a recursive function of integer variable n. Like all recursivefunctions, it has both recursive case and base case.

The portion of the definition that does not contain T is called the base case of the recurrence relation; the portion that contains T is called the recurrent or recursive case.
Recurrence relations are useful for expressing the running times (i.e., the number of basic operations executed) of recursive algorithm

The base case is reached when n == 0. The method performs one comparison.
Thus, the number of operations when n == 0, T(0), is some constant a.
When n > 0, the method performs two basic operations and then calls itself,
using ONE recursive call, with a parametern – 1.
Therefore the recurrence relationis:

```
void f(int n) { if (n
> 0) {
cout<<n; f(n-1);
} }
```

T(0) = a
T(n) = b + T(n-1)
where a is constant
where b is constant.

# DIVIDE AND CONQUER

General Method
In divide and conquer method, a given problem is,
i) Divided into smaller subproblems.
ii) These subproblems are solved independently.
iii) Combining all the solutions of subproblems into a solution of the whole.
If the subproblems are large enough then divide and conquer is reapplied. The generated subproblems
are usually of some type as the original problem.
Hence recursive algorithms are used in divide and conquer strategy.

```
Algorithm DAndC(P)
{
if small(P) then return
S(P)else{
divide P into smaller instances P1,P2,P3...Pk;
apply DAndC to each of these subprograms; // means DAndC(P1), DAndC(P2).....
DAndC(Pk)
return combine(DAndC(P1), DAndC(P2)..... DAndC(Pk));
}
}
//PProblem
//Here small(P) Boolean value function. If it is true, then the function S is
//invoked
```

Time Complexity of DAndC algorithm:
T(n) = T(1) if n=1
aT(n/b)+f(n) if n>1
a,b contants.
This is called the general divide and-conquer recurrence

Advantages of DAndC:
The time spent on executing the problem using DAndC is smaller than other method.
This technique is ideally suited for parallel computation.
This approach provides an efficient algorithm in computer science

Applications of Divide and conquer rule or algorithm:
Binary search,
Quick sort,
Merge sort,
Strassen's matrix multiplication