



# SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

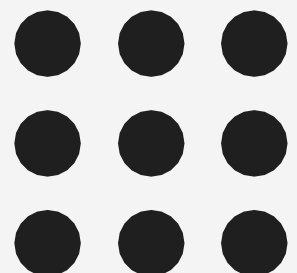
## Department of Artificial Intelligence and Data Science

### Course Name: 23ITB201 Data structures and Algorithms

II Year / III semester

Unit I – List ADT

Topic: Recursive and non recursive Algorithms





# Recursion and Non recursion



9/3/2024

SNSCE / AI&DS/AP/KALPANA C

2/10



# Recursion



## What is Recursion?

**Recursion** is a programming technique where a function calls **itself within its own definition**.

This allows a function to **break down a problem into smaller subproblems**, which are then solved recursively.

## What is a Recursive Algorithm?

A recursive algorithm is an algorithm that uses recursion to solve a problem. Recursive algorithms typically have two parts:

- 1. Base case:** Which is a condition that stops the recursion.
- 2. Recursive case:** Which is a call to the function itself with a smaller version of the problem.



# Examples of Recursion



## Examples of recursion:

- Example 1: Factorial: The factorial of a number  $n$  is the product of all the integers from 1 to  $n$ . The factorial of  $n$  can be defined recursively as:

$$\mathbf{factorial(n) = n * factorial(n-1)}$$

- Example 2: Fibonacci sequence: The Fibonacci sequence is a sequence of numbers where each number is the sum of the two preceding numbers.
- The Fibonacci sequence can be defined recursively as:

$$\mathbf{fib(n) = fib(n-1) + fib(n-2)}$$



# Non Recursive Algorithm



- Non Recursive Algorithm, also known as an iterative algorithm, involves solving a problem through repetition of a series of instructions until a specific condition is met, typically without the need for the function to call itself.
- Unlike recursive algorithms, non-recursive algorithms **do not involve function calls to itself**. Instead, they utilize **looping structures such as for-loops, while-loops, and do-while loops**, depending on the specific requirements of the problem and programming language.
- Each iteration repeats the same series of steps, manipulating the problem's input data until a solution is achieved.



# Example Non Recursive Algorithm



## Calculation of Factorial Number: .

- We make use of a loop to multiply the number with every other number smaller than it, until we reach the number 1.

```
int factorial(int num)
{
    int i=1,f=1;
    while(i<=num)
    {
        f=f*i;
        i++;
    }
    return f;
}
```



# Example Non Recursive Algorithm



## Calculation of Factorial Number: .

- We make use of a loop to multiply the number with every other number smaller than it, until we reach the number 1.

```
int factorial(int num)
{
    int i=1,f=1;
    while(i<=num)
    {
        f=f*i;
        i++;
    }
    return f;
}
```

