Date : 18th - 09 - 2020
Morning Session : 9am – 11.00 PM
By ~ Rohan Kumar

Topics: RECURSION - 1

Recursion: The term Recursion can be defined as the process of defining something in terms of itself. In simple words, it is a process in which a function calls itself directly or indirectly.

Syntax:

```
def func(): <--
              |
              | (recursive call)
              |
    func() ----
```

Advantages of using recursion

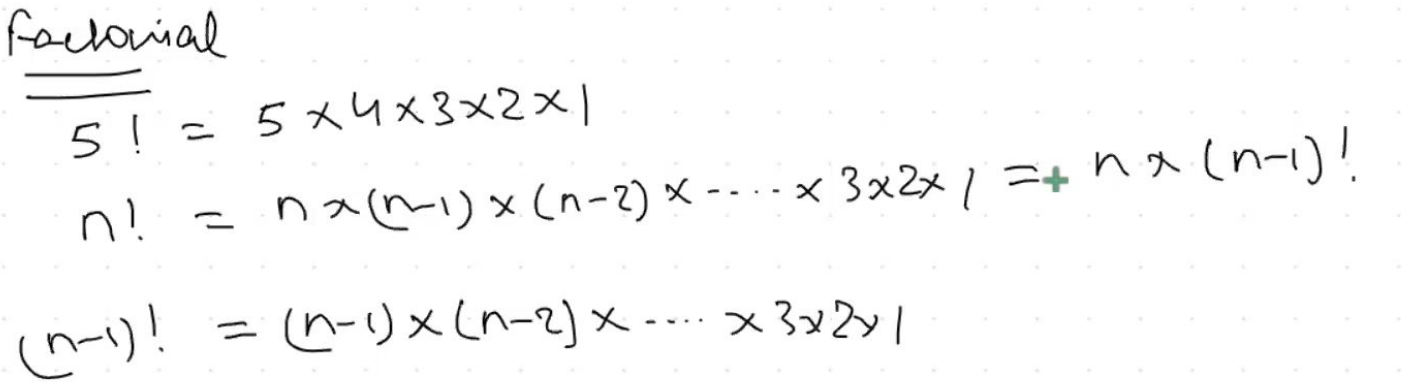
- A complicated function can be split down into smaller subproblems utilizing recursion.
- Sequence creation is simpler through recursion than utilizing any nested iteration.
- Recursive functions render the code look simple and effective.

Disadvantages of using recursion

- A lot of memory and time is taken through recursive calls which makes it expensive for use.
- Recursive functions are challenging to debug.
- The reasoning behind recursion can sometimes be tough to think through.

FACTORIAL: Factorial of a number specifies a product of all integers from 1 to that number. It is defined by the symbol explanation mark (!).

For example: The factorial of 5 is denoted as 5! = 1*2*3*4*5 = 120.



Example

```
def factorial(x):
    if x==1:
        return 1
    else:
        return x*factorial(x-1)

f=factorial(5)
print ("factorial of 5 is ",f)
```

Fibonacci: A Fibonacci sequence is a sequence of integers which first two terms are 0 and 1 and all other terms of the sequence are obtained by adding their preceding two numbers.

For example: 0, 1, 1, 2, 3, 5, 8, 13 and so on…

Fibonacci Resource

Time Complexity Of Fibonacci:

[Resource]

MCQ 1:

1. Recursion is always better than iterative approach
    a. True
    b. False

Answer: B; False

MCQ 2:

2. Which Time Complexity is better ?
    a. O(2 to the power n)
    b. O(n)

Answer: B; O(n)

MCQ 3:

3. Recursion always does not need Base case
    a. True
    b. False

Answer: B ; False

MCQ 4:

4. Recursion function
    a. keep calling simpler function of itself always
    b. Keep calling any function always

Answer: A; keep calling Simpler function of itself always