



TechM Full-Stack Software Development

Lecture On: Recursion

Instructor: Arkoprovo Dey

In Last Class, we covered....

- Mathematics



Homework Discussion

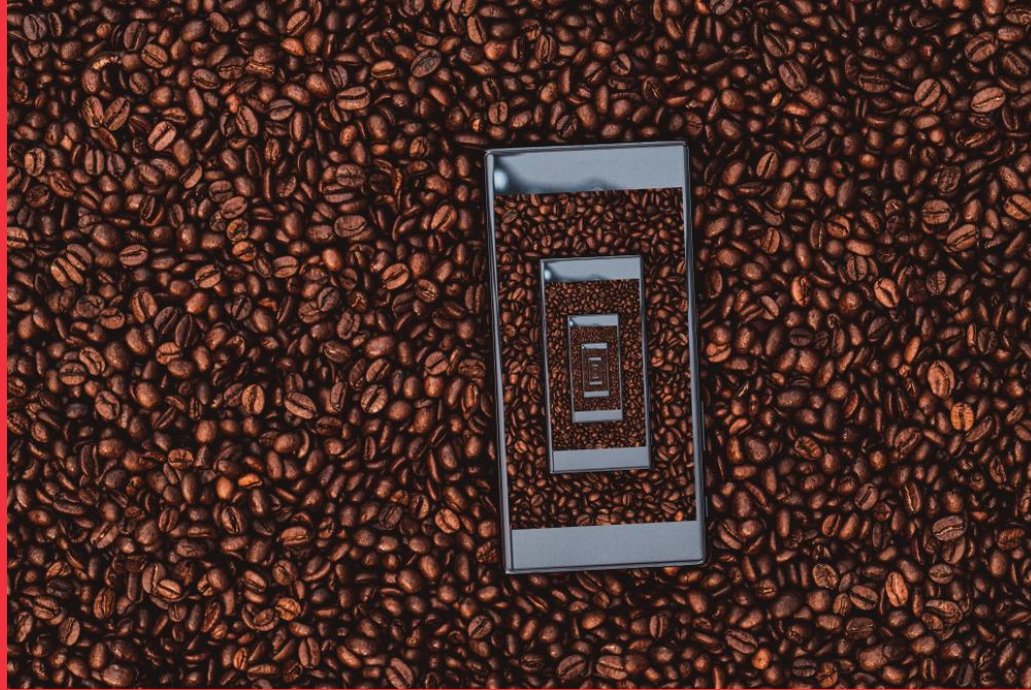
1. Write a program to reverse a input number.
2. Write a simple program which will take two numbers as input and prints the GCD and LCM of those two numbers.
3. Write a program which takes a number "n" as an input and prints first n fibonacci numbers.

Today's Agenda

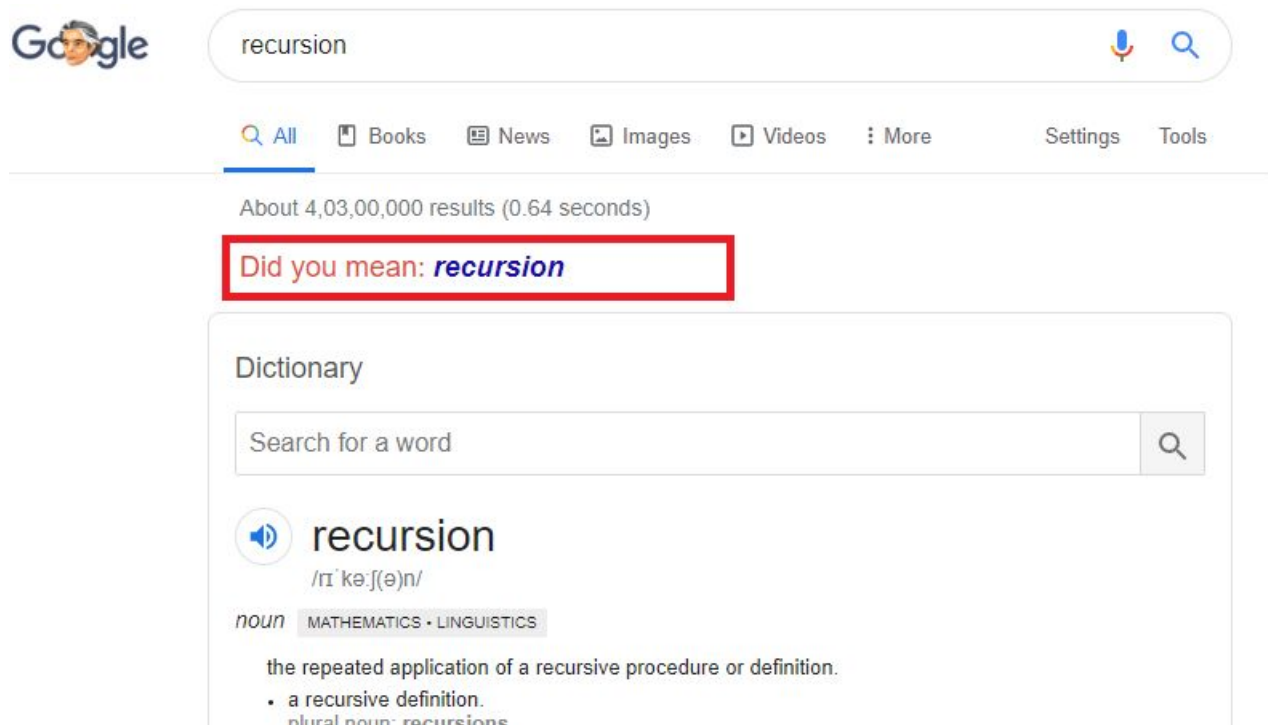
1 Recursion



Recursion



- If you search “recursion”, Google gives you a suggestion, “Did you mean: recursion”. When you click on the suggestion, it again loads the page and shows the same suggestion.



You might have heard about or seen Matryoshka, or Babushka, dolls. These are identical looking dolls stacked one inside the other. When you open up one doll, there is an identical one looking right back at you!



This is very similar to the concept of **Recursion**!

Recursion

Recursion provides us with an alternative to using recursive solutions. Every recursive solution can be implemented with the usage of loops iteratively. However, some of these solutions might have an inherent recursive design and might be tricky to design iteratively.

One must be very careful while going for a recursive solution. A function cannot keep calling itself forever and never get close to the solution. A recursive function might also end up doing the same computation multiple times, making it not as efficient as it can be.

Poll 1 (45 sec.)

Recursive solution can be implemented with the usage of loops iteratively.

1. True
2. False

Poll 1 (Answer)

Recursive solution can be implemented with the usage of loops iteratively.

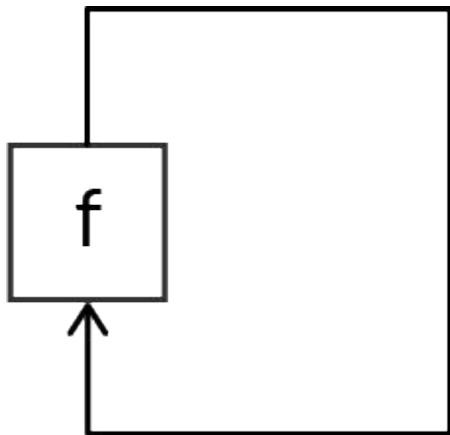
1. **True**

2. False

There are two forms of recursion or recursive calls:

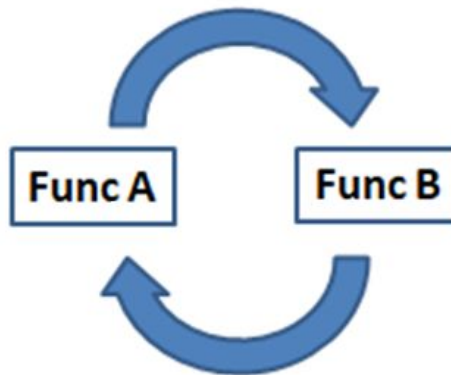
Direct Recursion

Where the recursive function calls itself explicitly.



Indirect Recursion

Where the recursive function does not call itself directly, but gets called via other functions.



Recursion

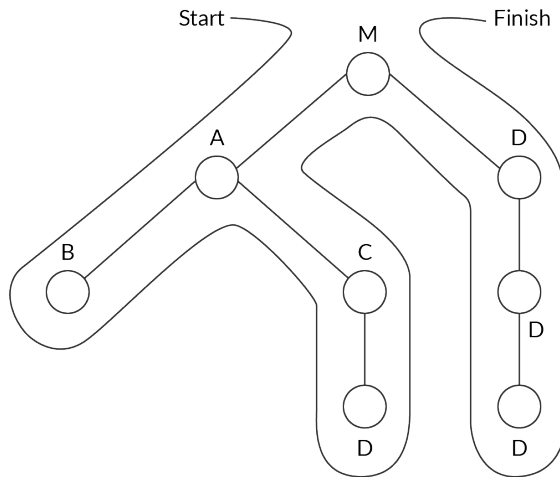
Every recursive function essentially has two parts:

- **The base case** – This is the condition where the inputs to the function are so trivial that we know the answer to it immediately. This is where the multiple recursive calls 'stop' and no more recursive calls are made. Every recursive solution **MUST** have a proper base case.
- **The recursive call** – The recursive call is the block of code that executes the method call to itself with the modified input. It also determines what should be done once the solution to the recursive call is made.

Once again, we should always make sure that the recursive calls always **'descend'** to the base case.

Recursion

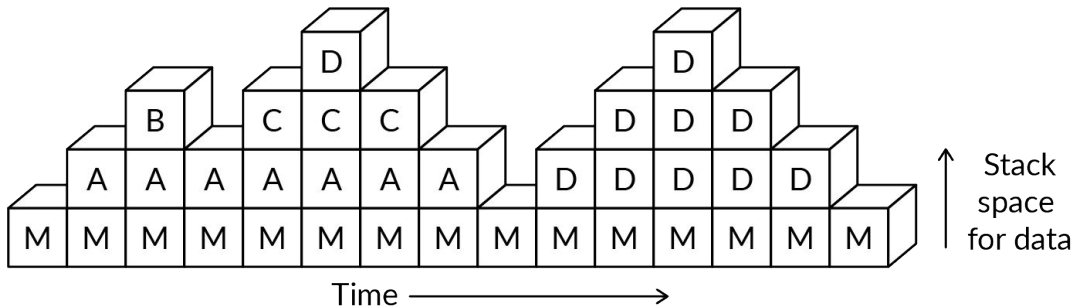
Recursive calls work just like any other function call. When a function calls another function, its execution is paused and its 'state' is captured and stored in the system Stack. When the function that was called has completed its work, the caller function resumes its computation.



Recursion

The same thing happens when a function calls itself. The state of the calling function is stored in the stack and the new function call is treated as a completely new function.

If we do not define a proper base case and the Stack keeps piling up, then our program might run out of stack space and lead to a **StackOverflow** error!



Poll 2 (45 sec.)

What is the space complexity of the adjoining code?

1. $O(1)$
2. $O(n)$
3. $O(n*n)$
4. None of these

```
static int fact(int n)
{
    if(n <= 1)
        return n;
    return n*fact(n-1);
}
```


Poll 2 (Answer)

What is the space complexity of the adjoining code?

1. $O(1)$
2. $O(n)$
3. $O(n*n)$
4. None of these

```
static int fact(int n)
{
    if(n <= 1)
        return n;
    return n*fact(n-1);
}
```

Poll 3 (45 sec.)

Which amongst the following options is true about recursion?

1. Recursive solution uses more memory than an iterative solution
2. Iterative solution uses more memory than a recursive solution
3. It is always better to use recursion whatsoever
4. None of these

Poll 3 (Answer)

Which amongst the following options is true about recursion?

- 1. Recursive solution uses more memory than an iterative solution**
2. Iterative solution uses more memory than a recursive solution
3. It is always better to use recursion whatsoever
4. None of these

Poll 4 (45 sec.)

What is the GCD of 10 and 15?

1. 2, 5
2. 10
3. 15
4. 5

Poll 4 (Answer)

What is the GCD of 10 and 15?

1. 2, 5
2. 10
3. 15
4. 5

Hands-on Coding

The greatest common divisor (gcd) of two numbers is the largest positive integer that divides each of the integers. For example, the gcd of 18 and 12 is 6.

Find GCD of two numbers.

Example:

Sample Input: 18 12

Sample Output: 6

Poll 5 (45 sec.)

What is the factorial of 3?

1. 3
2. 6
3. 10
4. 24

Poll 5 (Answer)

What is the factorial of 3?

1. 3
- 2. 6**
3. 10
4. 24

Hands-on Coding

Find factorial of a number.

Example:

Sample Input: 4

Sample Output: 24

Explanation: $(4! = 4*3*2*1 = 24)$

Poll 6 (45 sec.)

What would be the next number in the given fibonacci series?

0,1,1,2,3,5,8,13,21,34, ...

1. 40

2. 45

3. 55

4. 60

Poll 6 (Answer)

What would be the next number in the given fibonacci series?

0,1,1,2,3,5,8,13,21,34, ...

1. 40

2. 45

3. 55

4. 60

Hands-on Coding

The Fibonacci Sequence is the series of numbers:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

The next number is found by adding up the two previous numbers in the sequence.

You have to find the n^{th} term of the Fibonacci sequence.

Example:

Sample Input: 4

Sample Output: 3

Explanation: ($F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $F_4 = 3$)

Poll 7 (45 sec.)

What is the sum of digits of 123?

1. 4
2. 5
3. 6
4. 7

Poll 7 (Answer)

What is the sum of digits of 123?

1. 4

2. 5

3. 6

4. 7

Hands-on Coding

Find the sum of the digits in an integer.

Example:

Sample Input : 245

Sample Output : 11

Explanation: $(2+4+5 = 11)$

Homework

1. Write a program using recursion to check if the given number is a palindrome or not.
2. Write a program using recursion which checks if the number is divisible by 3 or not.

Tasks to complete after the session

Homework Questions
MCQs
Coding Questions

In the next class...

- We shall get introduced to a basic data structure: Arrays
- Solve some problems using Arrays





Thank You!