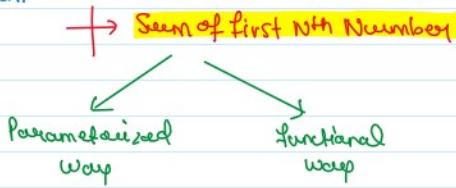


L3 Parameterized and Functional Recursion

Sunday, December 11, 2022 9:59 AM

In this Lecture we Learned what is the parameterized and function Revision with the problem



example

N=3

→ Next Means first (3) Number sum find next is any task

$$\text{ans} = \boxed{1+2+3} = 6$$

Parameterized Rewrite

Parameterized recursion are those recursion that function have some parameter

example = $f(n,m)$                         <img alt="red arrow pointing from the word 'function' to the parameter 'n'"



```

myfunction(N,sum){  

    if(i < 1){  

        sum = sum + i;  

    }  

    return sum;  

}

```

print(`myfunction(5, 0)`)
 └ function call
 for answer



Visualized and Revision

$$N=3, \text{Sum} = 0$$

```
myfunction (N,sum){  
    if (i < 1) {  
        print (sum)  
    }  
}
```

$$N=2, \text{Sum}=3$$

action ($N.sum$)
if ($i < 1$)
 print (sum)

$$N = 7, \text{Sum} = 5$$

function (N, sum) {
 if ($i < 1$) {
 return sum
 } else {
 return function (sum) {
 return N + sum;
 };
 }
}

81

N=3

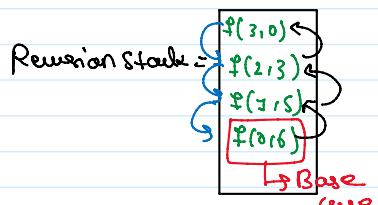
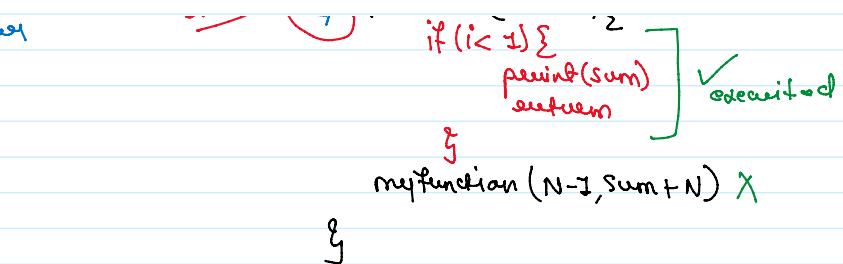
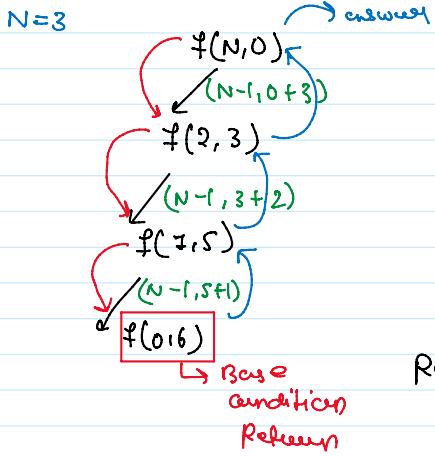
#RussianTree

$$\text{answer} \rightarrow f(N, 0)$$

gelen

$$N=0, \text{Sum}=6$$

$(N \text{ sum}) \{$
 $< 1\} \}$ 
penint(sum) ✓ execute it - ok
8:48 10:10:10



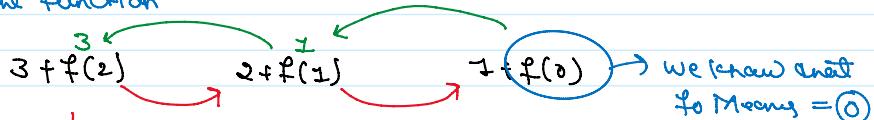
Time Complexity = $O(N)$
 ↗ Because we go made linearly

Space Complexity = $O(N)$
 ↗ Recursion Stack
 Space 

Functional Recursion

Functional recursion means we don't have a any parameter and we don't print anything just we return one function

Example



We know that to means = 0

→ that is the functional recursion

$N=3$

myfunction (N) {

if ($N == 0$) {
 return 0
}

return $N + \text{myfunction}(N-1)$

→ Functional Recursion
call 

Visualized That Recursion

$N=3$

myfunction (N) {

if ($N == 0$) {
 return 0
}

return $N + \text{myfunction}(N-1)$

$N=2$

myfunction (N) {

if ($N == 0$) {
 return 0
}

return $N + \text{myfunction}(N-1)$

$N=1$

myfunction (N) {

if ($N == 0$) {
 return 0
}

return $N + \text{myfunction}(N-1)$

$N=0$

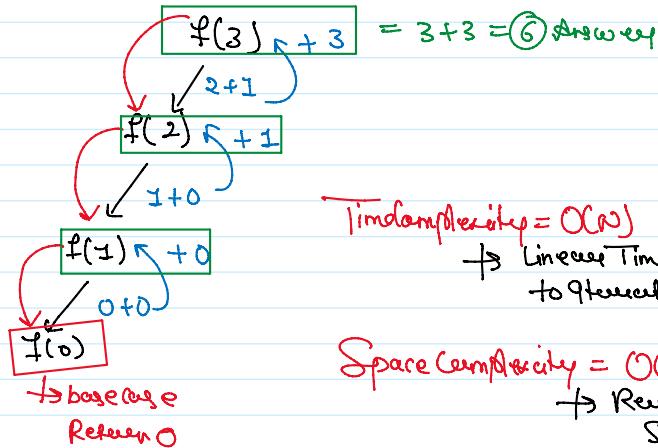
myfunction (N) {

if ($N == 0$) {
 return 0
}

return $N + \text{myfunction}(N-1)$

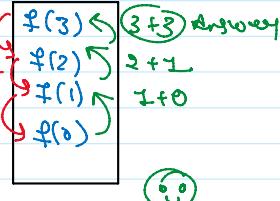
between $N + \text{mifunction}(N-i)$ ✓
 ↴ ↴ ↴ ↴ ↴
 3 f 3 2 + 1 1 + 0 X
 ↴ ↴ ↴ ↴
 ↴ what is our answer? 

#RussianTree



Space Complexity = $O(N)$

→ Russian Steak
Spaee



Implementation Python

```
# todo parametrized Function
```

```
def SumofN(N,sum):  
    if N == 0:  
        print(sum)  
        return  
  
    SumofN(N-1,sum+N)
```

```
# todo functional recursion
```

```
def SumofNth(N):  
    if N == 0:  
        return 0  
  
    return N + SumofNth(N-1)
```

```
print(SumofNth(5))
```

→ *parametrized Recursion*

→ *function Recursion*

1

#Implementation Java

```
public class L3_Parameterized_and_Functional_Recursion {  
    public static void main(String[] args) {  
  
        // todo parametrized Recursion  
  
        int n = 5;  
        SumofN(n,0);  
  
        // todo function recursion  
  
        int ans = SumofNth(n);  
        System.out.println(ans);  
    }  
  
    public static int SumofNth(int n) {  
  
        if(n == 0){  
            return 0;  
        }  
        return n + SumofNth(n-1);  
    }  
  
    public static void SumofN(int N,int sum){  
        if(N == 0){  
            System.out.println(sum);  
            return;  
        }  
        SumofN(N-1,sum+N);  
    }  
}
```