Recursion Application and Examples

Example-Fibonanci series

- **Fibonnacci Sequence**
- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
- Each element is the sum of the two preceding elements with

  fib(0) = 0

  fib(1) = 1

  fib(n) = n if n ==0 or n ==1

  fib(n) = fib(n-2) + fib(n-1) if n >= 2
- E.g.

Recursive Programming

- **Fibonnacci Sequence (0,1,1,2,3,5,8,13,…..)**

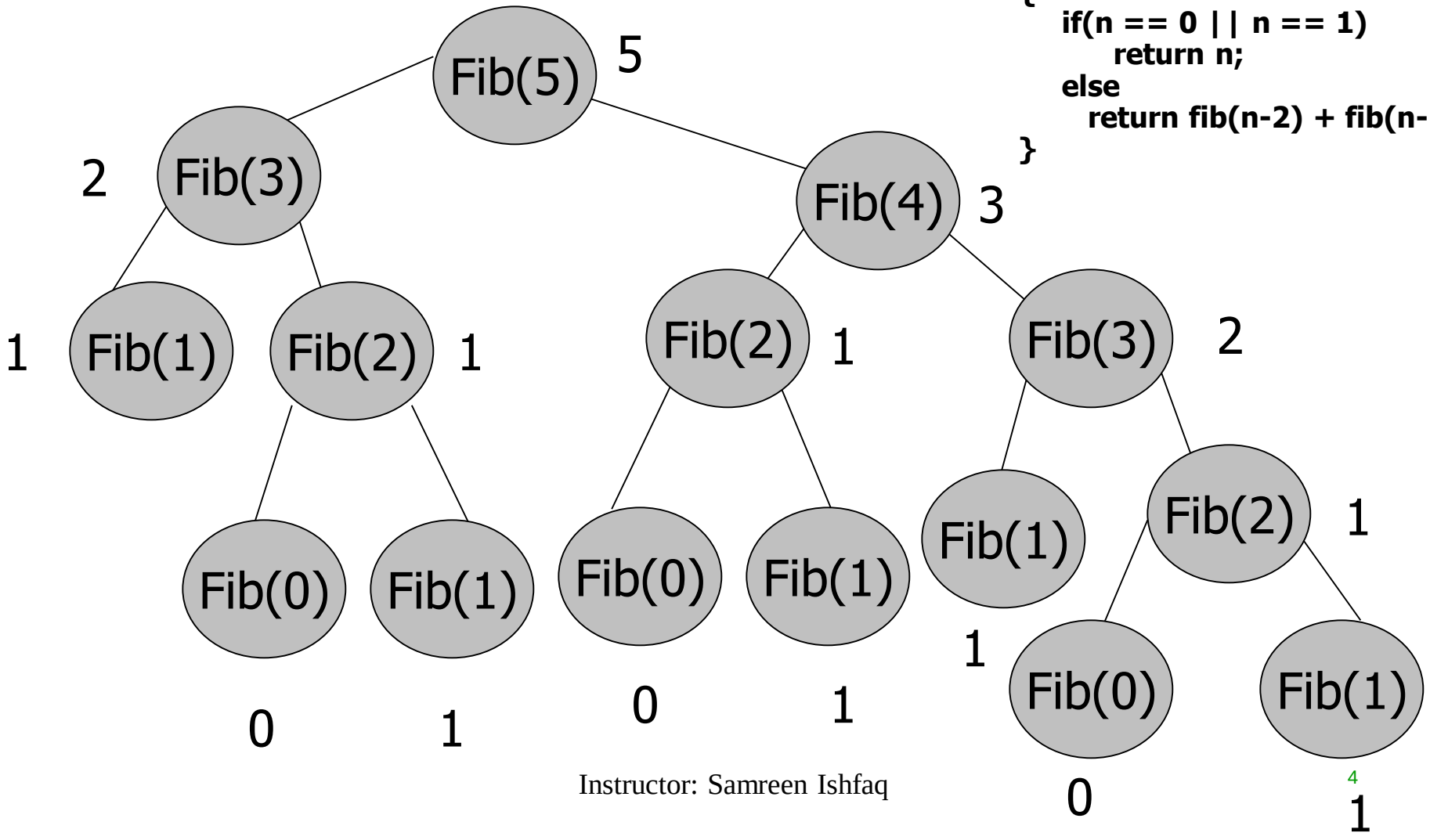
fib(n) = n if n ==0 or n ==1

fib(n) = fib(n-2) + fib(n-1) if n >= 2

```
int fib(int n)
{
    if(n == 0 || n == 1)
        return n;
    else
        return fib(n-2) + fib(n-1)
}
```

Problem To Solved (Fib(5))

```
int fib(int n)
{
    if(n == 0 || n == 1)
        return n;
    else
      return fib(n-2) + fib(n-1);
}
```



Dry Run (Home Task)

- Find fib(7) and how many function calls were executed?

Example – Find

Home Task

- To find an element in an array
- Base case
  - If array is empty, return false
- Recursive step
  - If 1st element of array is given value, return true
  - Skip 1st element and recur on remainder of array

Example – Count

Home Task

- To count # of elements in an array
- Base case
  - If array is empty, return 0
- Recursive step
  - Skip 1st element and recur on remainder of array
  - Add 1 to result

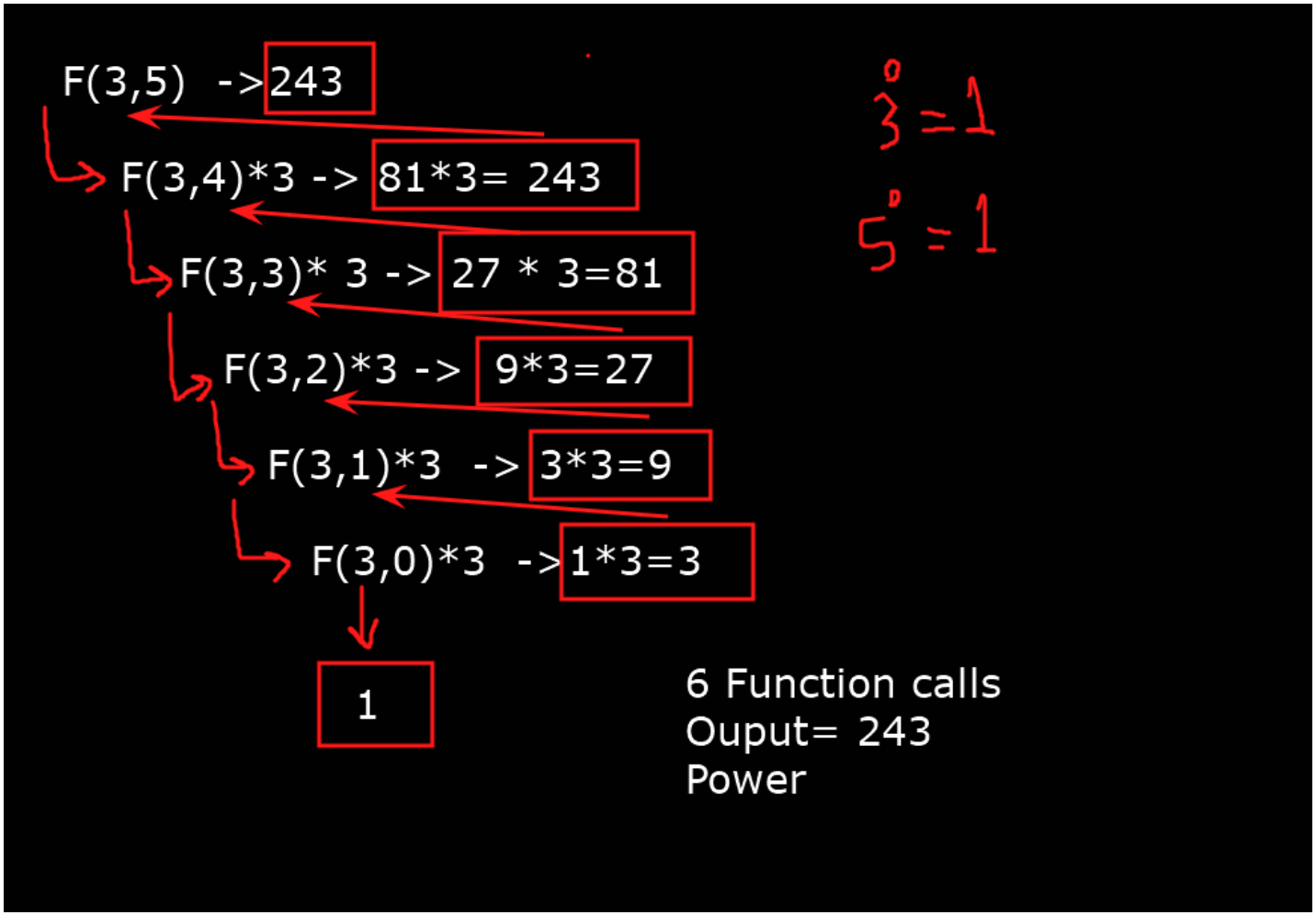
Example 1:

Input: x=3, n=5

```
int F(int x,int n)
{
        if(n==0)
                return 1;
        else
                return F(x,n-1)*x;
}
```

1. **Show complete Dry Run.**
2. **What does this function do/compute?**
3. **What will be the output of this function?**
4. **How many functions calls are there?**

Solution:


F(3,5) ->243
F(3,4)*3 -> 81*3= 243
F(3,3)* 3 -> 27 * 3=81
F(3,2)*3 -> 9*3=27
F(3,1)*3 -> 3*3=9
F(3,0)*3 -> 1*3=3
1
3^0 = 1
5^0 = 1
6 Function calls
Ouput= 243
Power

Example 2:

Input: N=5

```
int F(int n)
{
        if(n==0)
                return 0;
        else
                return n+F(n-1);
}
```

1. **Show complete Dry Run.**
2. **What does this function do/compute?**
3. **What will be the output of this function?**
4. **How many functions calls are there?**

Example 3:

Input:

```
int f(char c1,char c2 )
{
 if(c1>c2)
        return 0;
 if(c1+1==c2)
        return 1;
   return f(c1+1,c2-1)+2;
}
Function call with :
F('a','e'); & F('h','c');
```

1. **Show complete Dry Run.**
2. **What does this function do/compute?**
3. **What will be the output of this function?**
4. **How many functions calls are there?**

Example 4:

```
void F(char ch)
{
        if('A'<=ch && ch <='H')
        {
            F(ch-1)
            cout<<ch;
        }
        else
        {
            cout<<endl;
        }
}
```

Input: F('G')

1. **Show complete Dry Run.**
2. **What does this function do/compute?**
3. **What will be the output of this function?**
4. **How many functions calls are there?**

Example 5:

```
void F(char ch)
{
        if('A'<=ch && ch <='H')
        {
            F(ch+1)
            cout<<ch;
        }
        else
        {
            cout<<endl;
        }
}
```

Input: F('C')

1. **Show complete Dry Run.**
2. **What does this function do/compute?**
3. **What will be the output of this function?**
4. **How many functions calls are there?**

Example 6:

```
void F(char ch)
{
        if('A'<=ch && ch <='H')
        {

            cout<<ch;
             F(ch-1)
        }
        else
       {
            cout<<endl;
      }
}
```

Input: F('C')

1. **Show complete Dry Run.**
2. **What does this function do/compute?**
3. **What will be the output of this function?**
4. **How many functions calls are there?**

Example 7:

```
void F(char ch)
{
        if('A'<=ch && ch <='H')
        {

            cout<<ch;
             F(ch-1)
        }
        else
       {
            cout<<endl;
      }
}
```

Input: F('3')

1. **Show complete Dry Run.**
2. **What does this function do/compute?**
3. **What will be the output of this function?**
4. **How many functions calls are there?**