

A job ready bootcamp in C++, DSA and IOT

Recursion



Saurabh Shukla (MySirG)

Agenda

- ① What is a recursion?
- ② Recursion Tree | Tracing code
- ③ How to approach recursive solution?
- ④ Few examples

What is a Recursion?

- Function calling itself is called recursion.
- A recursive method solves a problem by calling a copy of itself to work on a smaller problem.
- It is important to ensure that the recursion terminates.

- Each time the function call itself with a slightly simpler version of the original problem.
- Recursive code is generally shorter and easier to write than iterative code.
- Solution to some problems are easier to formulate recursively.

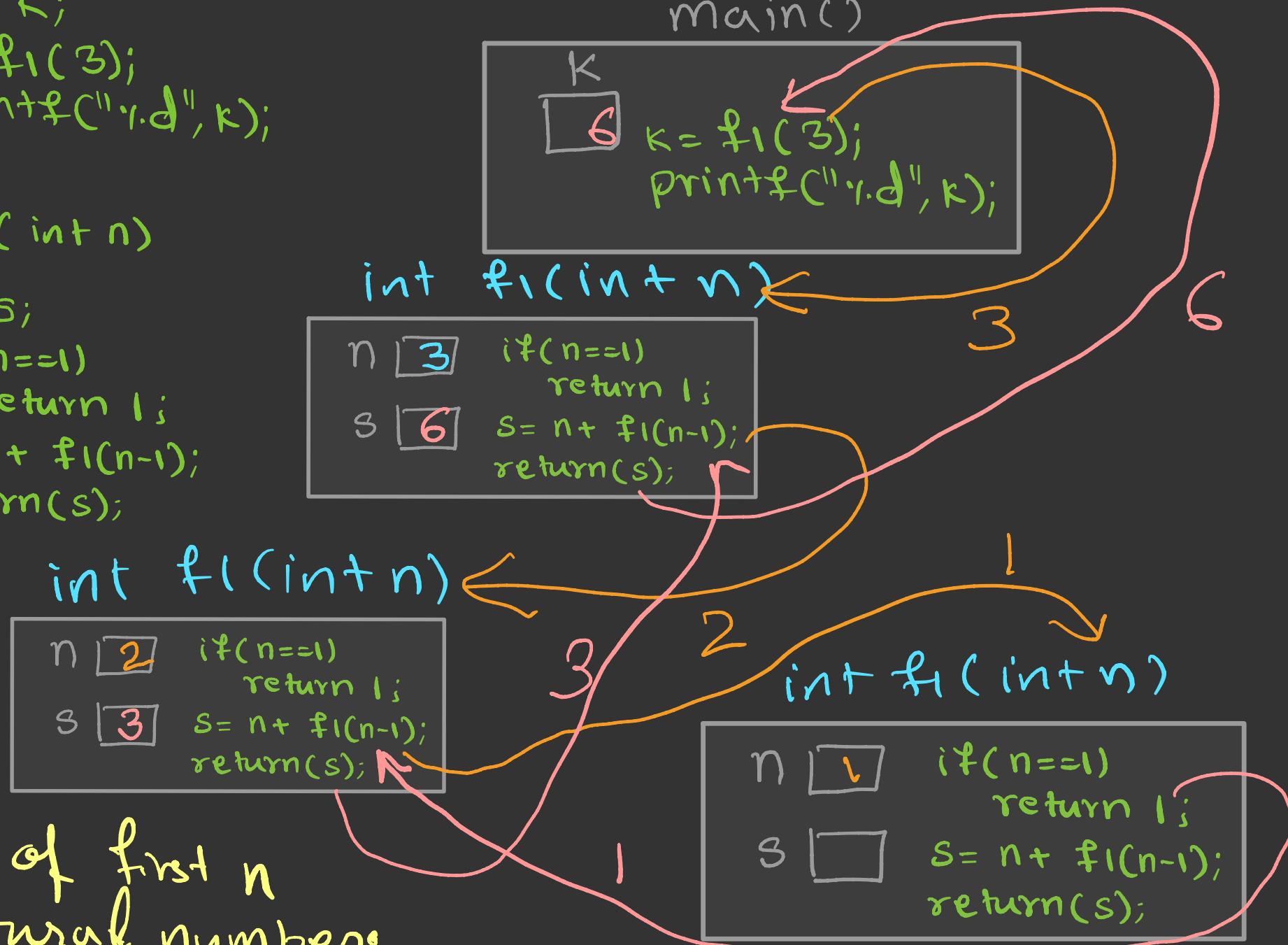
```

int main()
{
    int K;
    K = f1(3);
    printf("%d", K);
}

int f1(int n)
{
    int s;
    if(n==1)
        return 1;
    s = n + f1(n-1);
    return(s);
}

```

Recursion Tree



How to approach a Recursive Problem?

Write a recursive function to calculate sum of first n natural numbers.

```
int sum(int n)
```

```
{
```

```
    if(n==1)
```

```
        return 1;
```

```
    return n + sum(n-1);
```

```
}
```

① $\text{sum}(n) = 1+2+3+4+\dots+n$

RC ② $n + \text{sum}(n-1) = 1+2+3\dots+n-1$

BC ③
n >= 1 recursion
n = 0 recursion steps

n == 1 return 1

$$\text{Sum}(n) = 1+2+3+4+\dots+n-1 + n$$

$$\text{Sum}(n-1) = \underline{1+2+3+4+\dots+n-1}$$

15 sum(5);

 \u2191

 5 + sum(4)

 \u2191 10

 4 + sum(3)

 \u2191 6

 3 + sum(2)

 \u2191 3

 2 + sum(1)

 \u2191 1

return 1

→ n == 1

return n + sum(n-1) → otherwise

Write a recursive function to calculate factorial of n.

int fact(int n)

{

 if(n==0 || n==1)
 return 1;

 return n*fact(n-1);

}

①

RC ②

BC ③

fact(n) $1 \times 2 \times 3 \times \dots \times n$

$n \times \text{fact}(n-1) \quad 1 \times 2 \times 3 \times \dots \times (n-1)$

0! or 1!

$n! \quad n \times (n-1)!$

Write a recursive function to print first n natural numbers.

```
void printN(int n) {  
    if(n>0)  
    {  
        printN(n-1);  
        printf("%d", n);  
    }  
}
```

} {

$\text{RC } \textcircled{1}$ $\text{printN}(n) \ 1\ 2\ 3\ 4\ ... \ n$

$\text{RC } \textcircled{2}$ $\left[\begin{array}{l} -\text{printN}(n-1), 1\ 2\ 3\ 4\ ... \ n-1 \\ -\text{printf}("%d", n); \end{array} \right]$

$\text{BC } \textcircled{3}$ $\left\{ \begin{array}{ll} n==0 & \text{BC} \\ n>0 & \text{RC} \end{array} \right.$