Full stack web development using python

Recursion

Saurabh Shukla (MySirG)

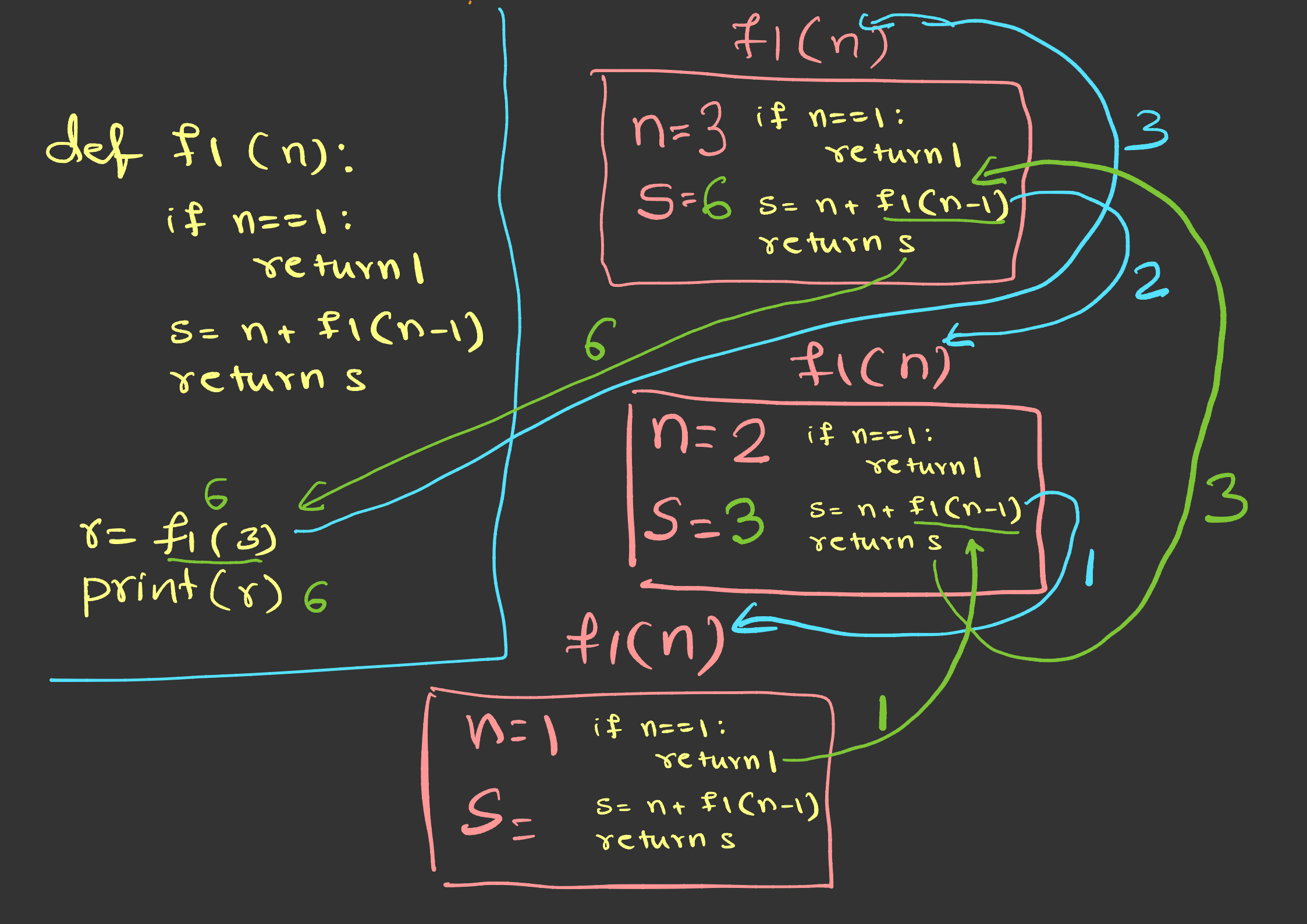
Agenda

1. What is recursion?
2. Recursion Tree
3. How to approach recursive solution?

Recursion

Function calling itself is called recursion.

```
def f1():
    ==
    f1()
    ==
```


def f1(n):
if n==1:
return 1
s= n+ f1(n-1)
return s
r= f1(3)
print(r)
6
f1(n)
n=3
s=6
if n==1:
return 1
s= n+ f1(n-1)
return s
f1(n)
n=2
s=3
if n==1:
return 1
s= n+ f1(n-1)
return s
f1(n)
n=1
s=
if n==1:
return 1
s= n+ f1(n-1)
return s
3
2
1
1
3
6
6

f1(5)	15	s = 5 + f1(4)	5+4+3+2+1
f1(4)	10	s = 4 + f1(3)	4+3+2+1
f1(3)	6	s = 3 + f1(2)	3+2+1
f1(2)	3	s = 2 + f1(1)	2+1
f1(1)	1	return 1	1

$$s = n + f1(n-1) \Rightarrow n+(n-1)+(n-2)+\ldots+4+3+2+1$$

n=100

100 + f1(99)

↓

99 + f1(98)

↓

98 + f1(97)

- In recursion, problem is solved in terms of problem itself
- Each time recursive function call to itself for little simpler version of the original problem

How to approach recursive function?

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

① f1(n) 1+2+3+...+n

RC ② n+f1(n-1) 1+2+3+...+(n-1)

BC ③ n==1 1

```
def f1(n):
    if n==1:
        return 1
    return n+f1(n-1)
```

$$f_1(n) = \begin{cases} n + f1(n-1) & n > 1 \\ 1 & n == 1 \end{cases}$$