n! Done Recursively

Matt Bishop

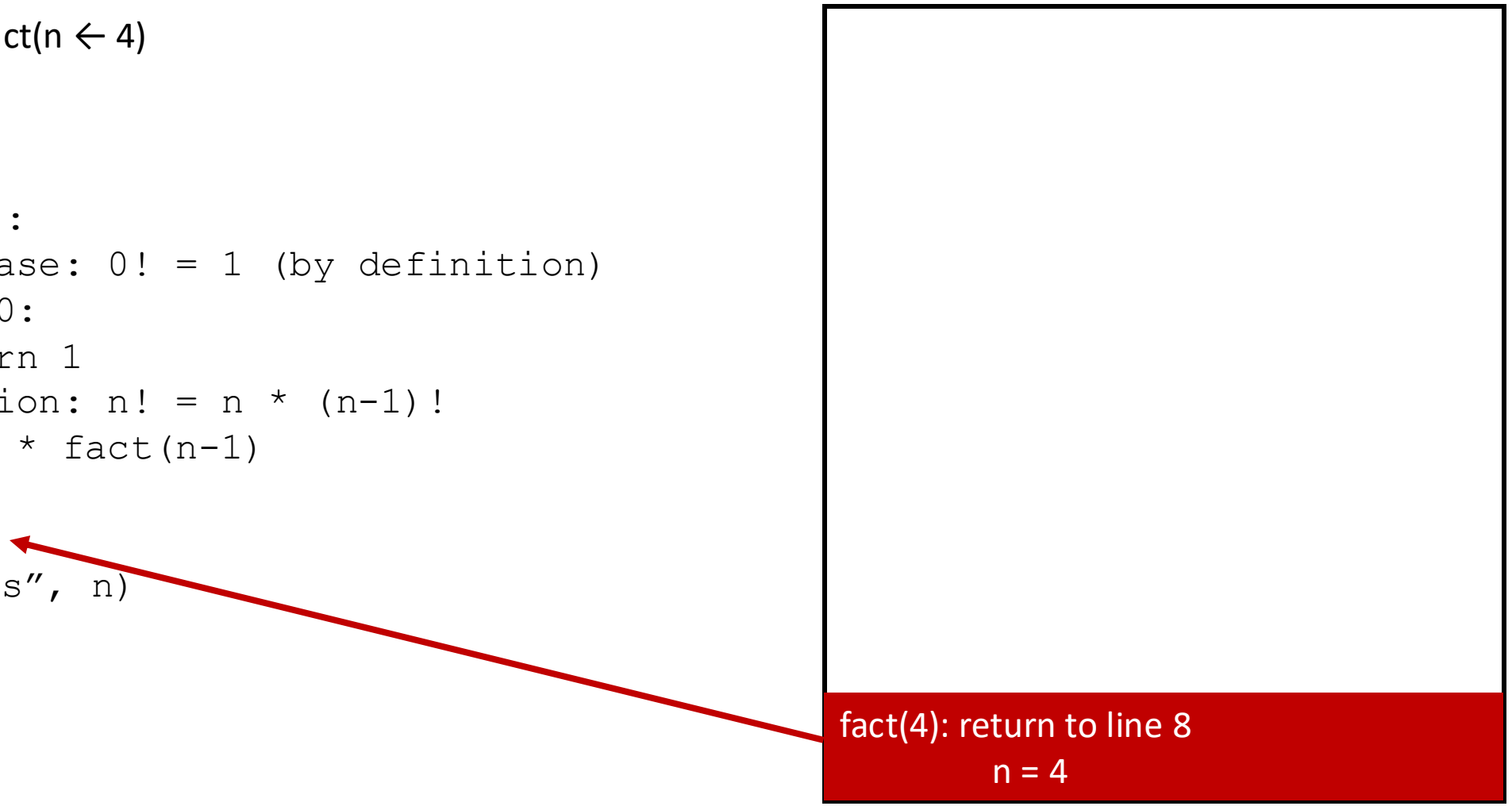
MHI 289I, Fall Quarter 2025

The Recursive n! Program

```
1: def fact(n):
2:       # base case: 0! = 1 (by definition)
3:       if n == 0:
4:           return 1
5:       # recursion: n! = n * (n-1)!
6:       return n * fact(n-1)
7:
8: n = fact(4)
9: print(“4! is”, n)
```

Initial call to fact: fact(n ← 4)

```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```

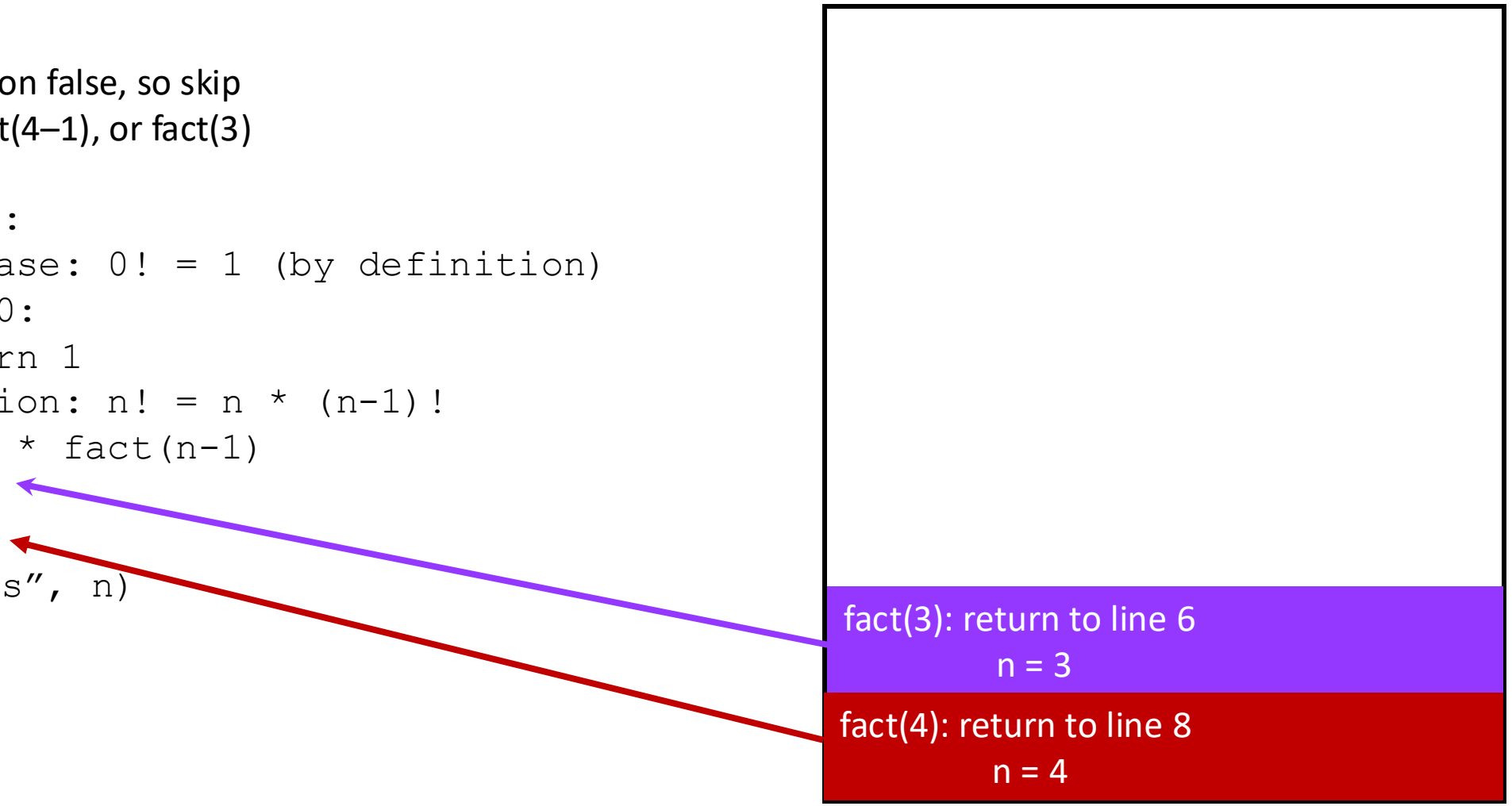


fact(n ← 4):

3: condition false, so skip

6: call fact(4–1), or fact(3)

```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```

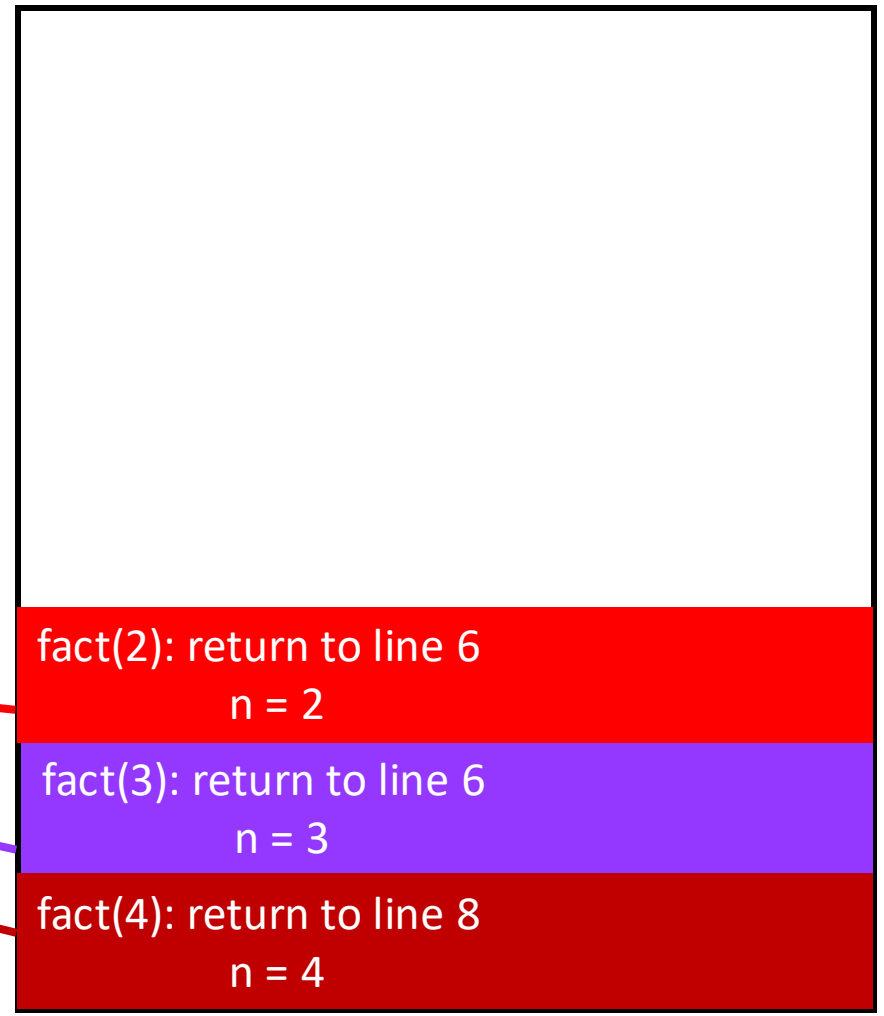


fact(n ← 3):

3: condition false, so skip

6: call fact(3–1), or fact(2)

```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```

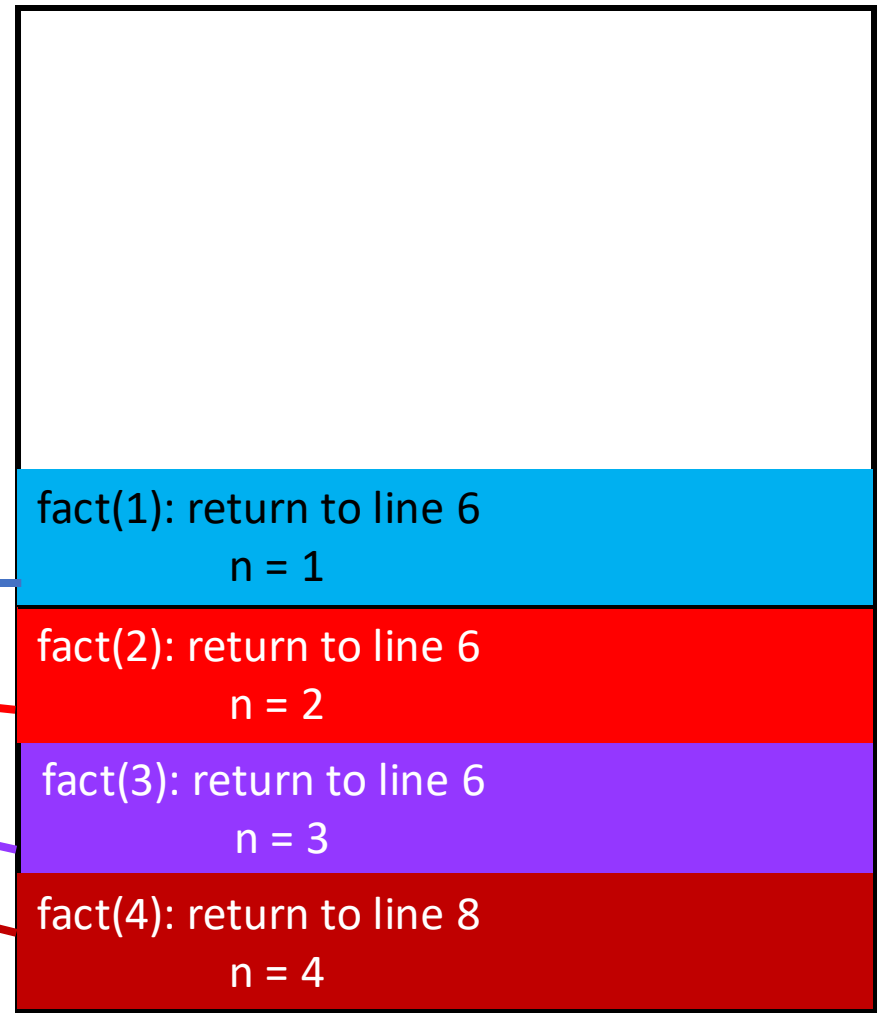


fact(n ← 2):

3: condition false, so skip

6: call fact(2–1), or fact(1)

```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```

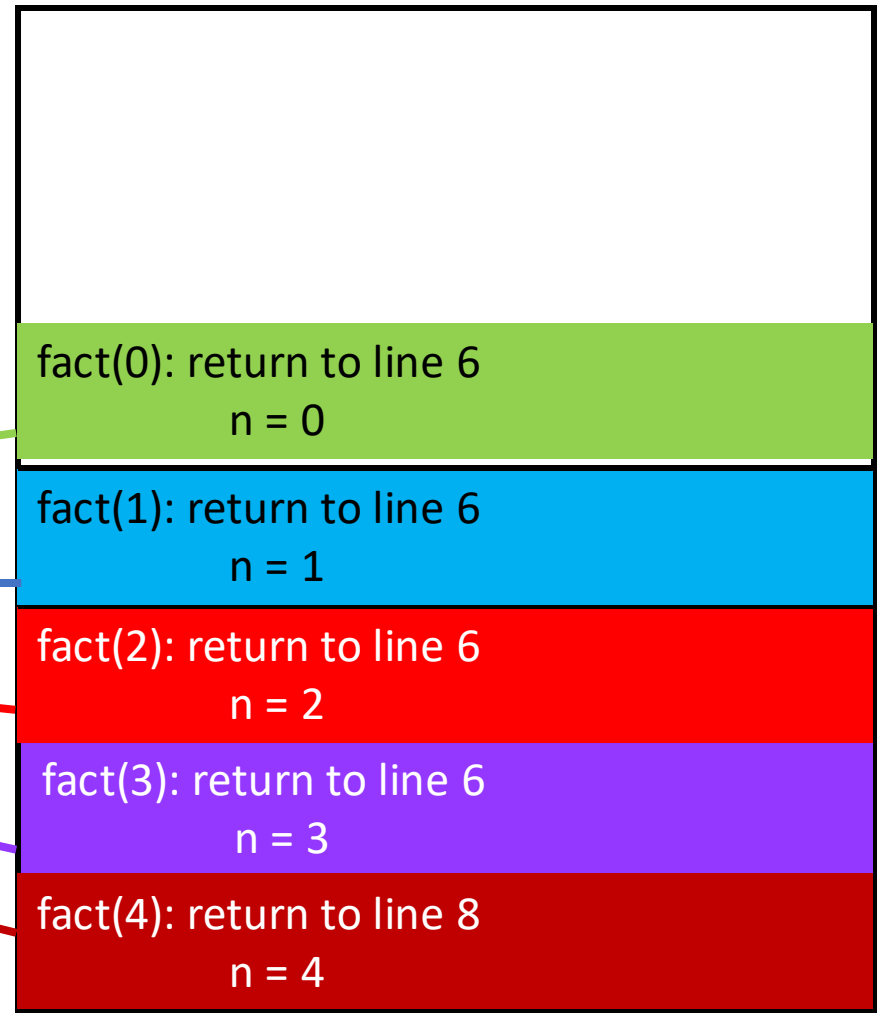


fact(n ← 1):

3: condition false, so skip

6: call fact(1–1), or fact(0)

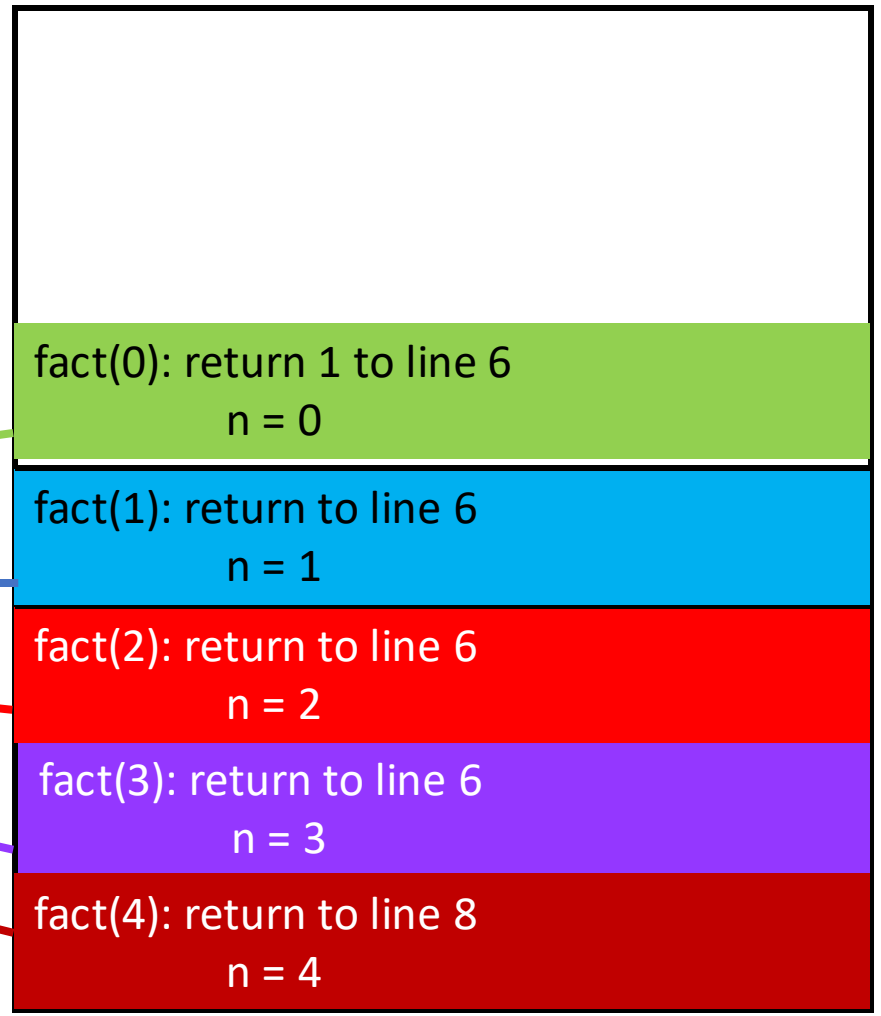
```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```



fact(n ← 0):

6: condition true, so return 1

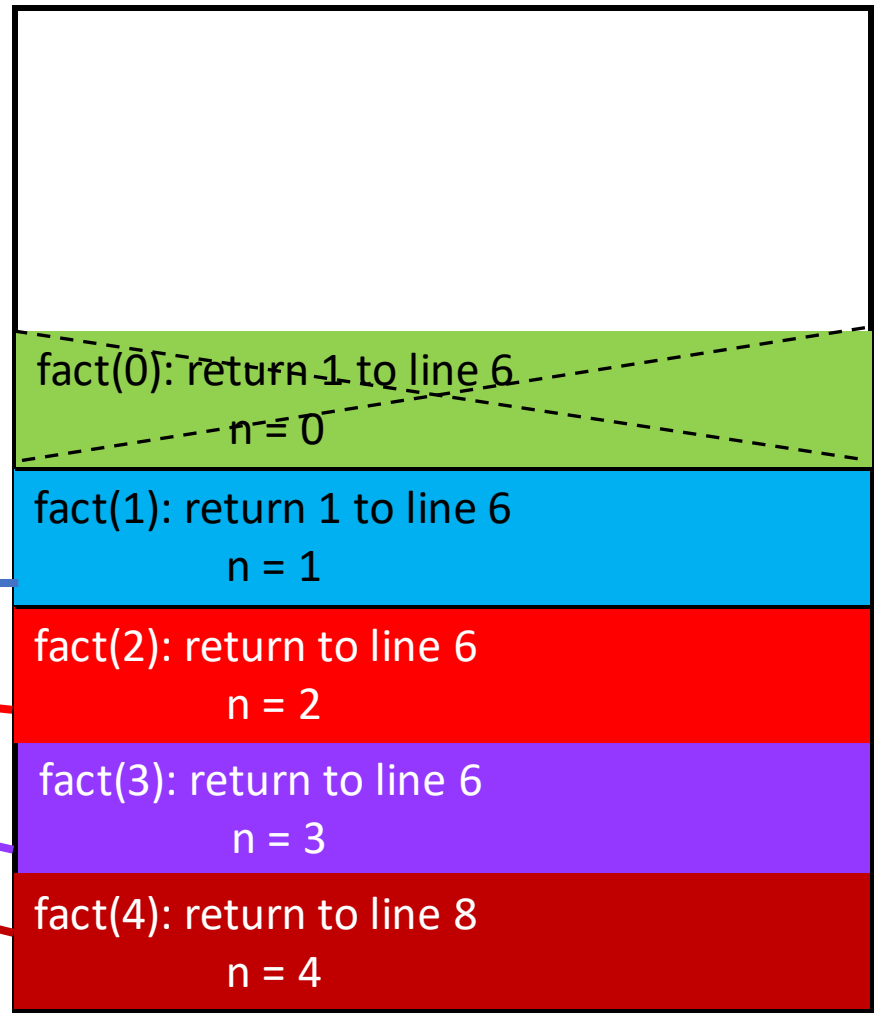
```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```



fact(n ← 1):

6: fact(0) = 1, so return 1 × 1 = 1

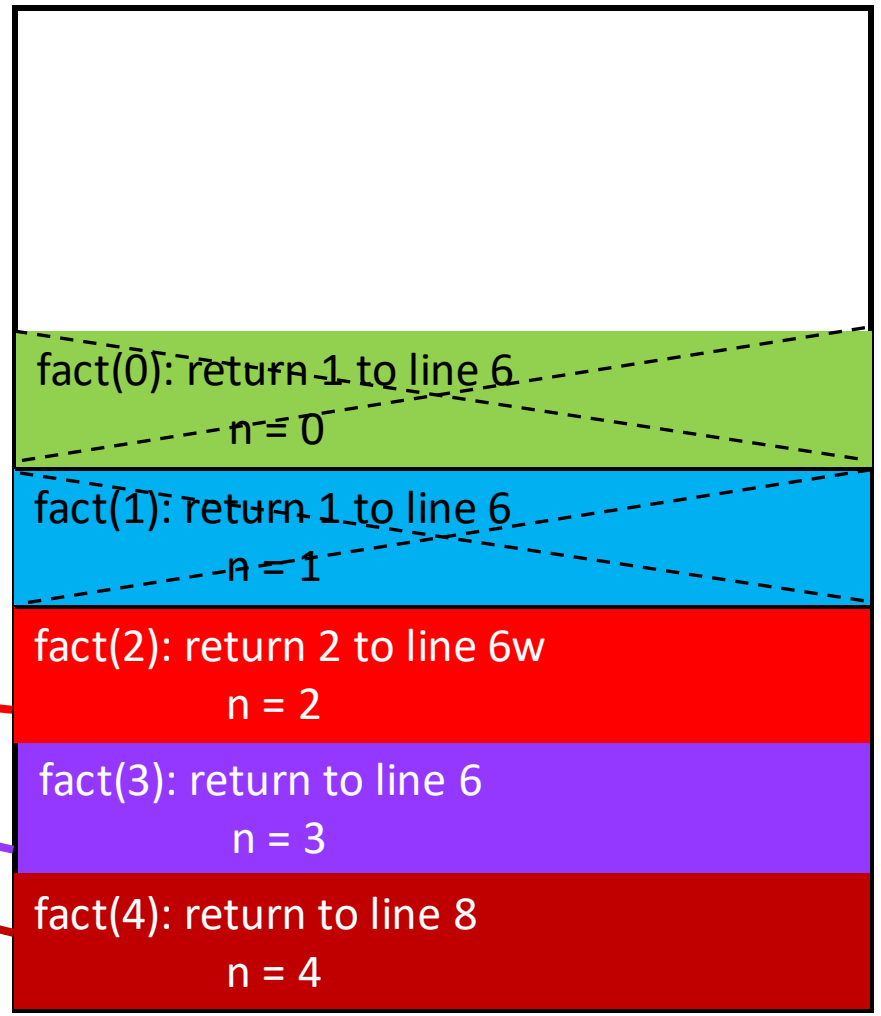
```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```



fact(n ← 2):

6: fact(1) = 1, so return 2 × 1 = 2

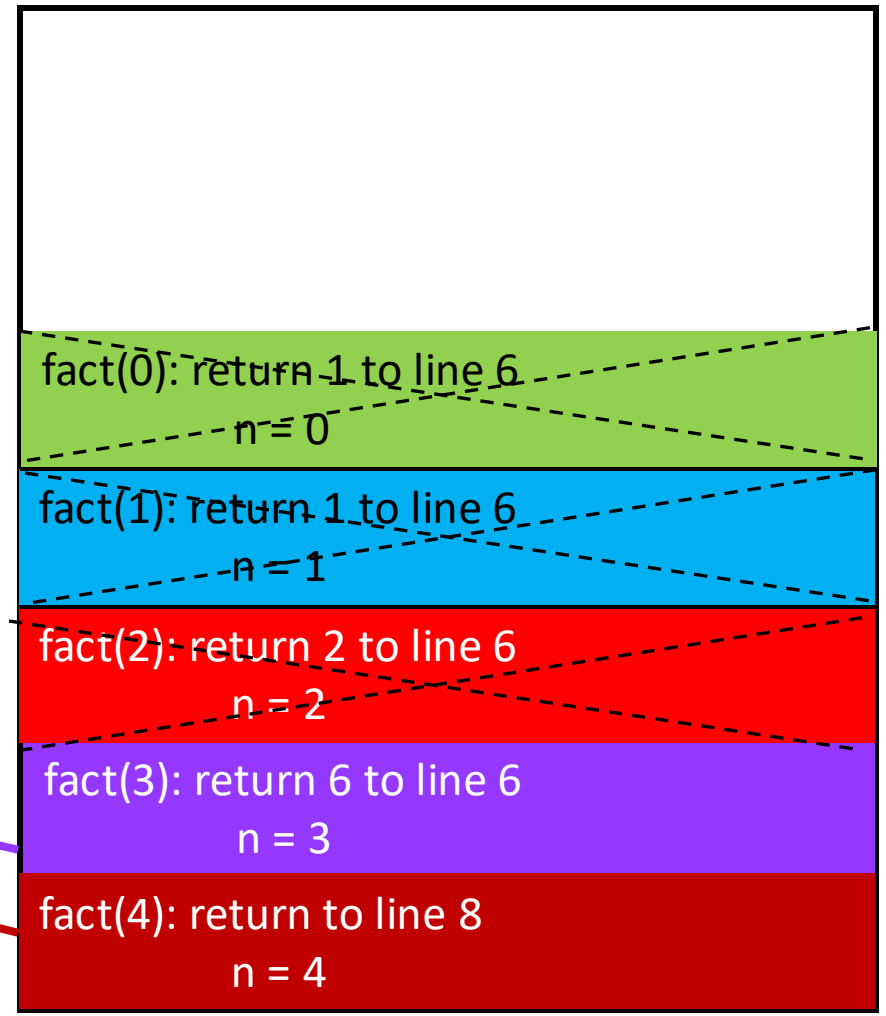
```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```



fact(n ← 3):

6: fact(2) = 2, so return 3 × 2 = 6

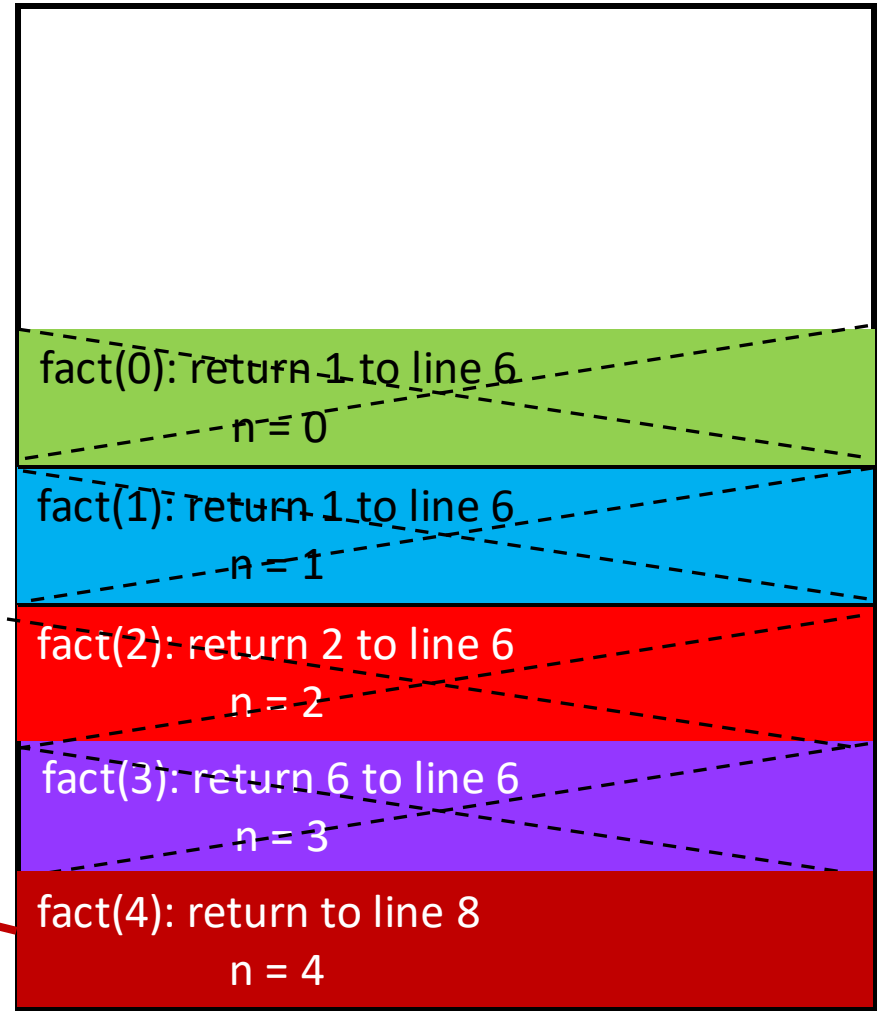
```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```



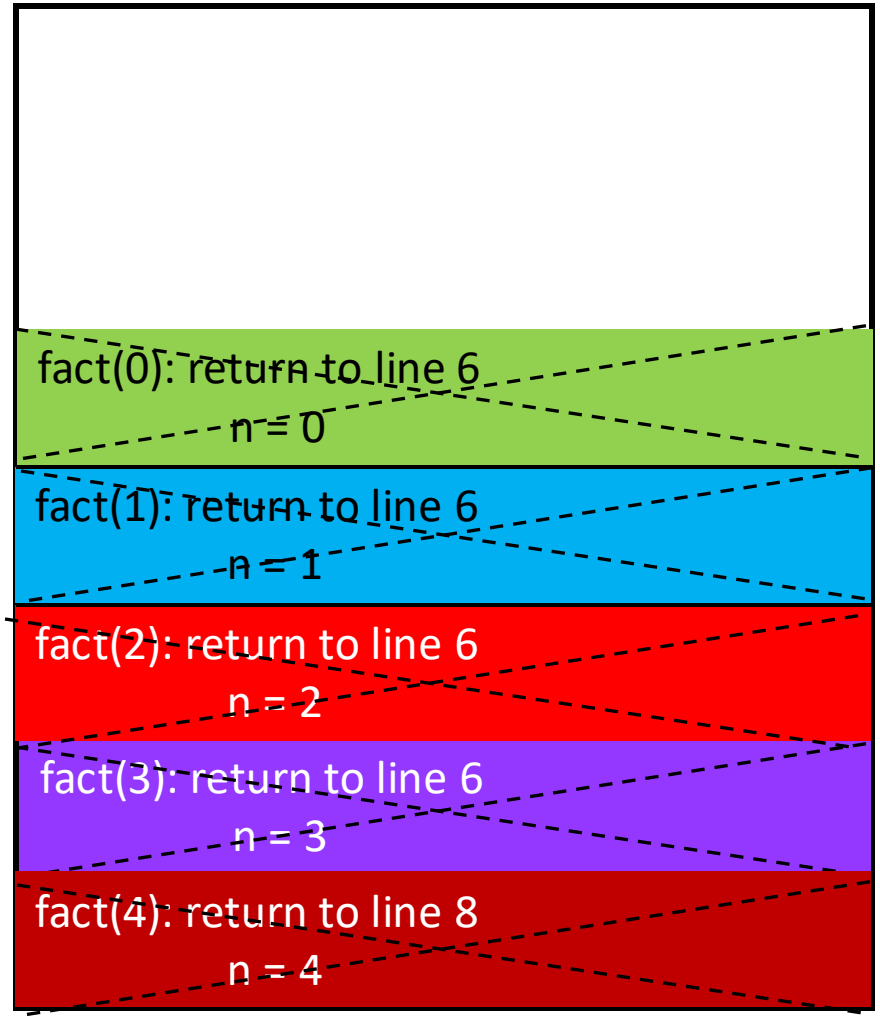
fact(n ← 4):

6: fact(3) = 6, so return 4 × 6 = 24

```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```



```
1: def fact(n):
2:     # base case: 0! = 1 (by definition)
3:     if n == 0:
4:         return 1
5:     # recursion: n! = n * (n-1)!
6:     return n * fact(n-1)
7:
8: n = fact(4)
9: print("4! is", n)
```

Printed:

4! is 24