Chapter 17: Recursion

Sections 17.1 – 17.2

Textbooks: Y. Daniel Liang, Introduction to Programming with C++, 3rd Edition

These slides were adapted by Prof. Gheith Abandah from the Computer Engineering Department of the University of Jordan for the Course: Computer Skills for Engineers (0907101)

Outline

- Introduction
- Example: Factorials

Motivations

- *Recursion is a technique that leads to elegant solutions to problems that are difficult to program using simple loops.*
- *A recursive function is one that invokes itself.*
- Suppose you want to find all the files under a directory that contains a particular word. How do you solve this problem? There are several ways to solve this problem. An intuitive solution is to use recursion by searching the files in the subdirectories recursively.

Outline

- Introduction
- Example: Factorials

Factorials in Math

0! = 1	-
1! = 1	1 x 0!
2! = 2 x1	2 x 1!
3! = 3 x 2 x 1	3 x 2!
4! = 4 x 3 x 2 x 1	4 x 3!
5! = 5 x 4 x 3 x 2 x 1	5 x 4!
6! = 6 x 5 x 4 x 3 x 2 x 1	6 x 5!
7! = 7 x 6 x 5 x 4 x 3 x 2 x 1	7 x 6!
8! = 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1	8 x 7!
9! = 9 x 8 x 7 x 6 x5 x 4 x 3 x 2 x 1	9 x 8!
10! = 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1	10 x 9!

Computing Factorial

$n! = n \times (n - 1) \times (n - 2) \times ... \times 2 \times 1$

$0! = 1;$

$n! = n \times (n - 1)!; n > 0$

$factorial(0) = 1;$

$factorial(n) = n*factorial(n-1);$

ComputeFactorial

Run

ComputeFactorial.cpp

```cpp
#include <iostream>
using namespace std;

// Return the factorial for a specified index
long long factorial(int n)
{
    if (n == 0) // Base case
        return 1;
    else
        return n * factorial(n - 1); // Recursive call
}

int main()
{
    // Prompt the user to enter an integer
    cout << "Please enter a non-negative integer: ";
    int n;
    cin >> n;

    // Display factorial
    cout << "Factorial of " << n << " is " << factorial(n);

    return 0;
}
```

```
Factorial of 4 is 24
```

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4)

animation

Computing Factorial

factorial(4) = 4 * factorial(3)

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4) = 4 * factorial(3)

= 4 * 3 * factorial(2)

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4) = 4 * factorial(3)

= 4 * 3 * factorial(2)

= 4 * 3 * (2 * factorial(1))

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4) = 4 * factorial(3)

= 4 * 3 * factorial(2)

= 4 * 3 * (2 * factorial(1))

= 4 * 3 * (2 * (1 * factorial(0)))

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4) = 4 * factorial(3)

= 4 * 3 * factorial(2)

= 4 * 3 * (2 * factorial(1))

= 4 * 3 * (2 * (1 * factorial(0)))

= 4 * 3 * (2 * (1 * 1)))

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4) = 4 * factorial(3)

= 4 * 3 * factorial(2)

= 4 * 3 * (2 * factorial(1))

= 4 * 3 * (2 * (1 * factorial(0)))

= 4 * 3 * (2 * (1 * 1)))

= 4 * 3 * (2 * 1)

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4) = 4 * factorial(3)

= 4 * 3 * factorial(2)

= 4 * 3 * (2 * factorial(1))

= 4 * 3 * (2 * (1 * factorial(0)))

= 4 * 3 * (2 * (1 * 1)))

= 4 * 3 * (2 * 1)

= 4 * 3 * 2

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4) = 4 * factorial(3)

= 4 * 3 * factorial(2)

= 4 * 3 * (2 * factorial(1))

= 4 * 3 * (2 * (1 * factorial(0)))

= 4 * 3 * (2 * (1 * 1)))

= 4 * 3 * (2 * 1)

= 4 * 3 * 2

= 4 * 6

animation

Computing Factorial

factorial(0) = 1;

factorial(n) = n*factorial(n-1);

factorial(4) = 4 * factorial(3)

= 4 * 3 * factorial(2)

= 4 * 3 * (2 * factorial(1))

= 4 * 3 * (2 * (1 * factorial(0)))

= 4 * 3 * (2 * (1 * 1)))

= 4 * 3 * (2 * 1)

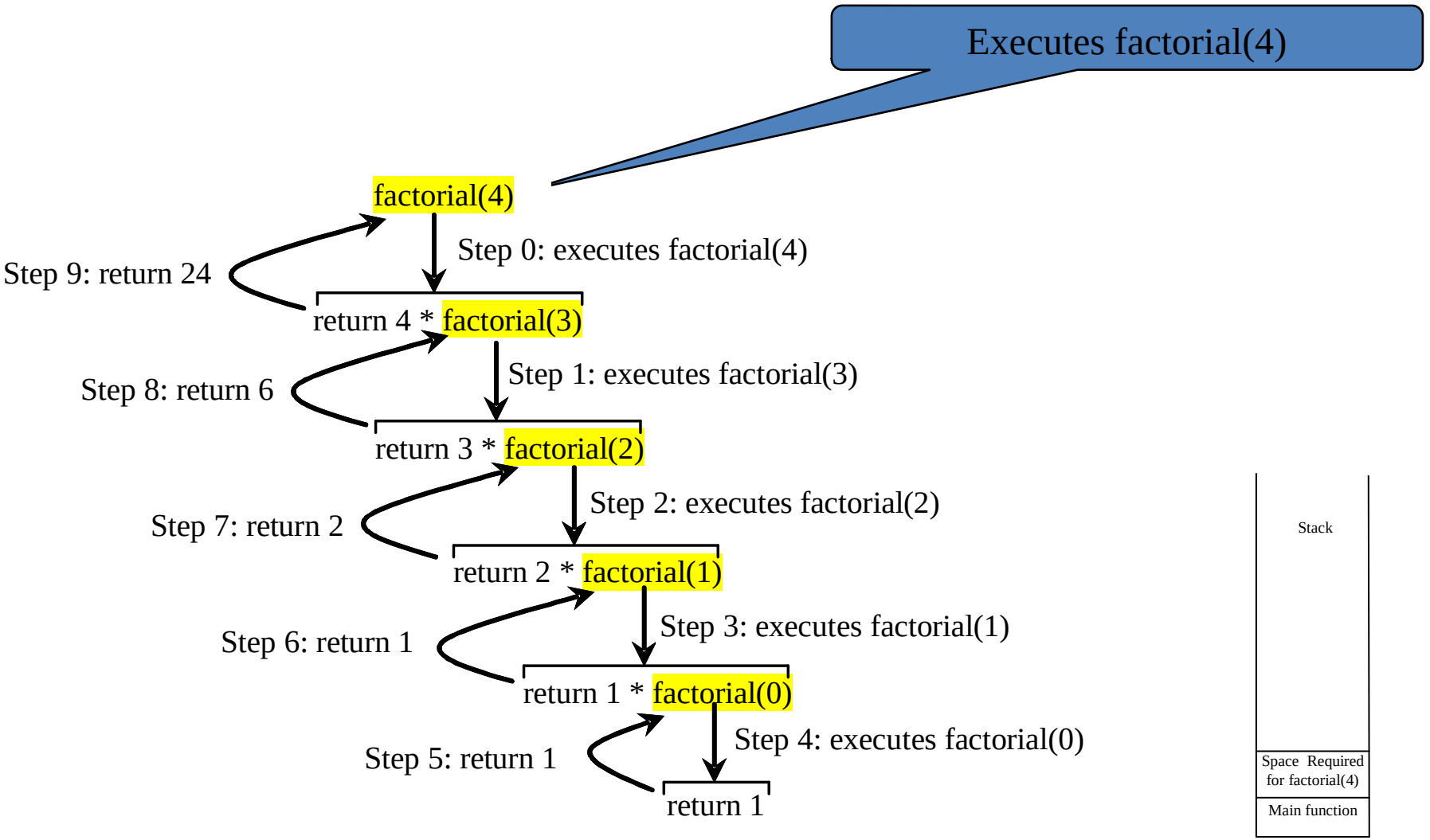
= 4 * 3 * 2

= 4 * 6

= 24

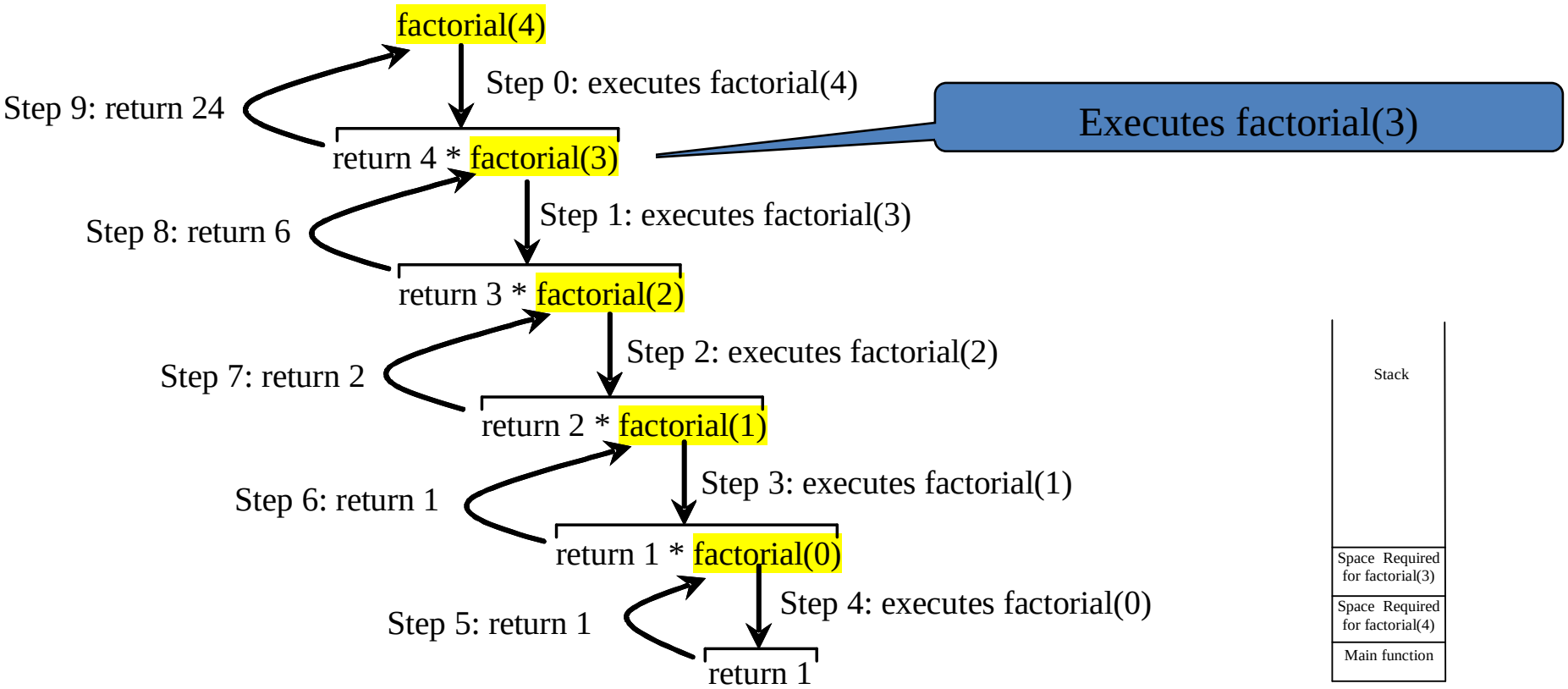
animation

Trace Recursive factorial



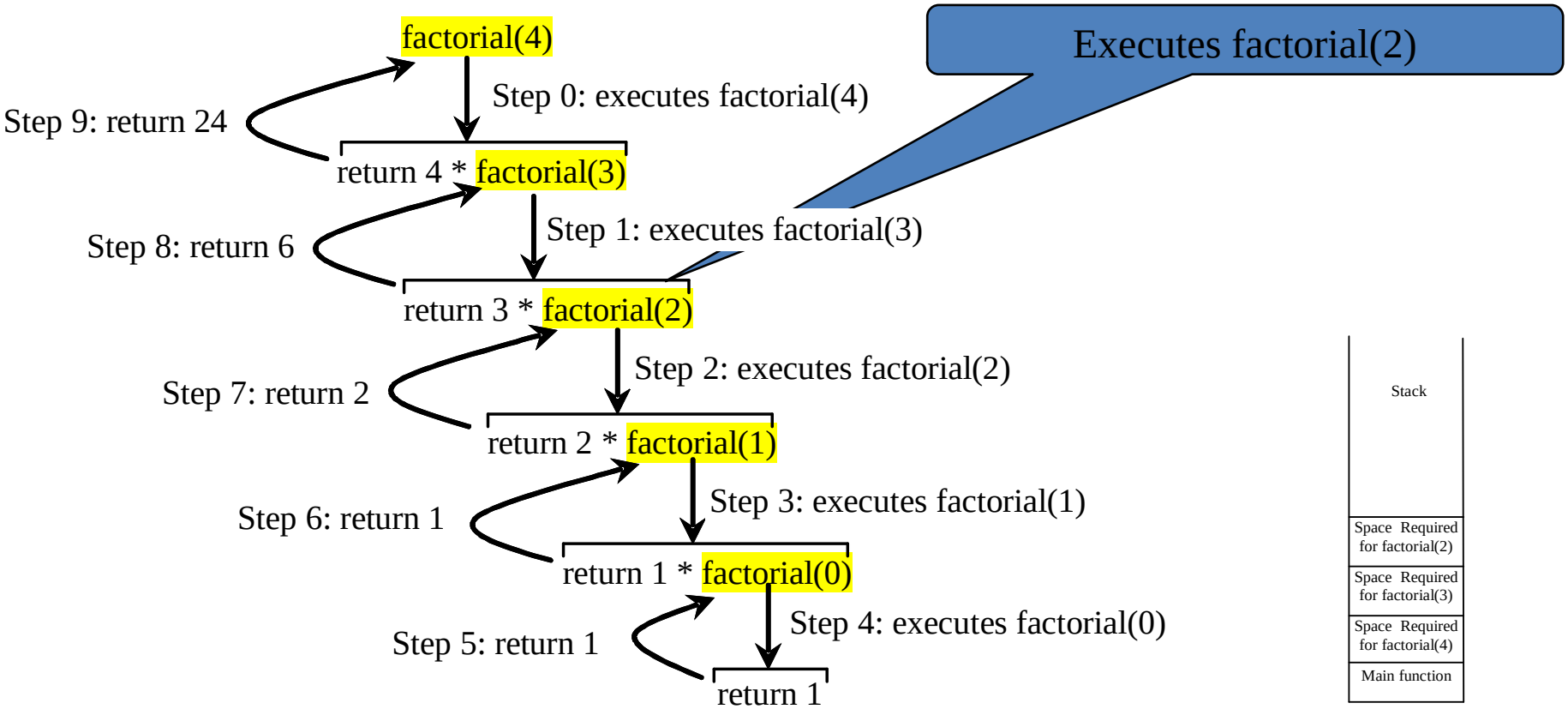
animation

Trace Recursive factorial



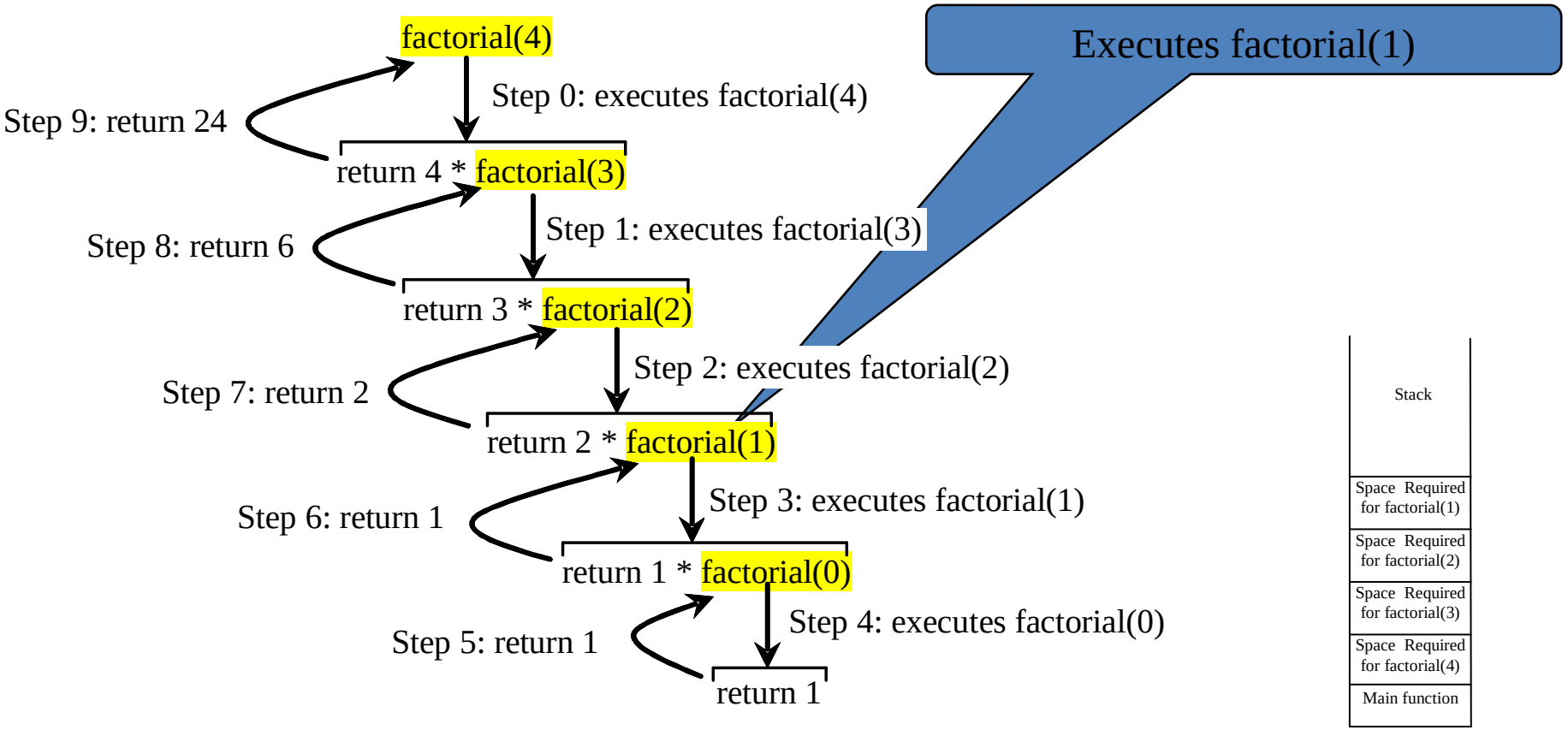
animation

Trace Recursive factorial



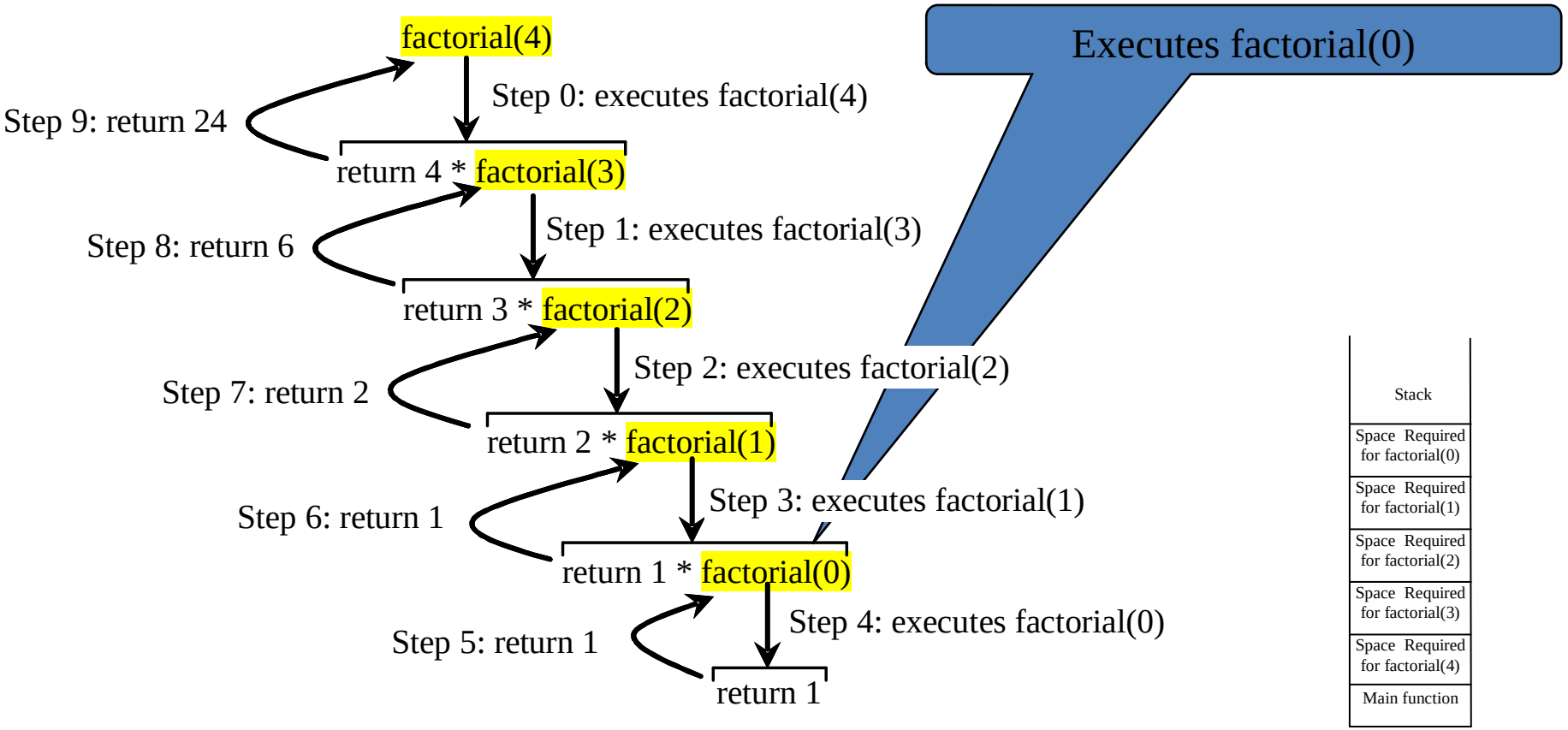
animation

Trace Recursive factorial



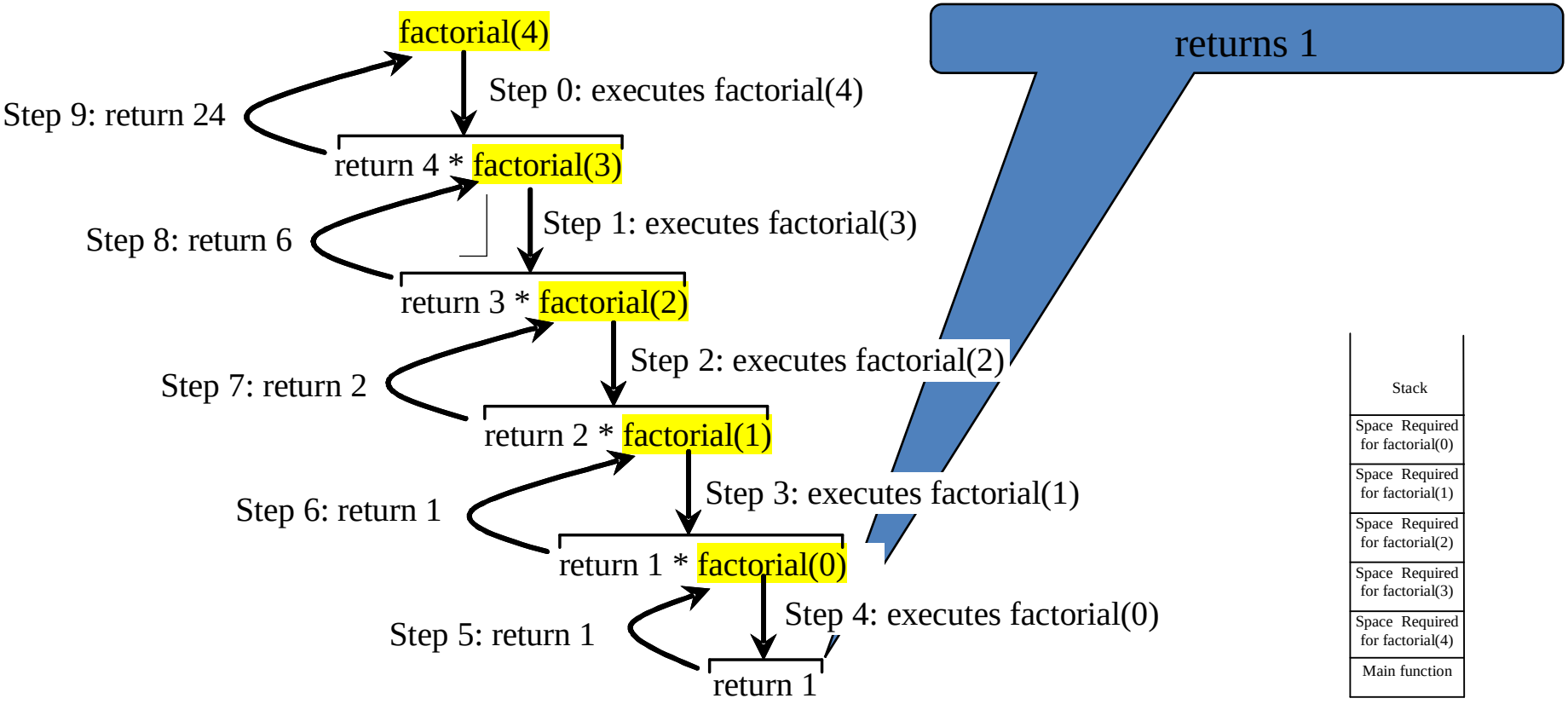
animation

Trace Recursive factorial



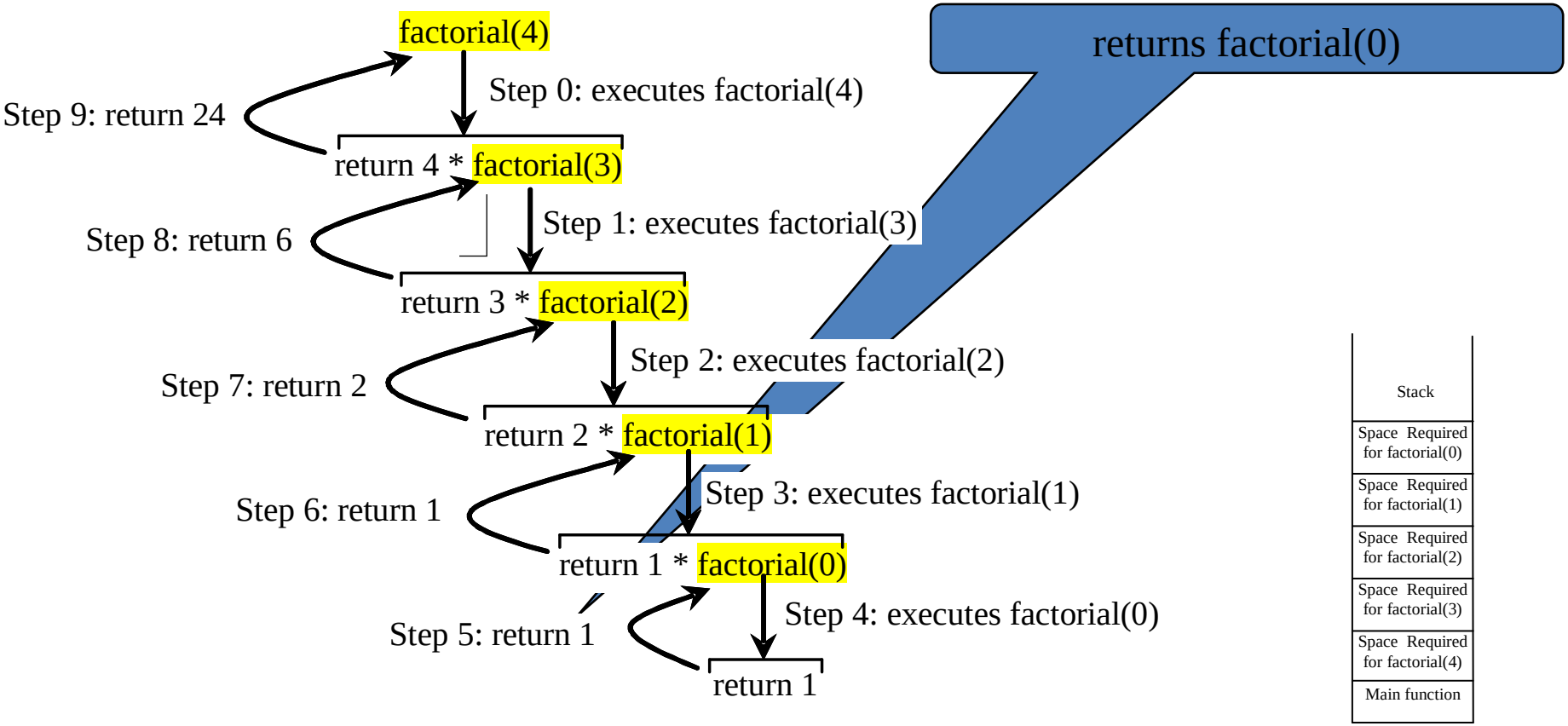
animation

Trace Recursive factorial



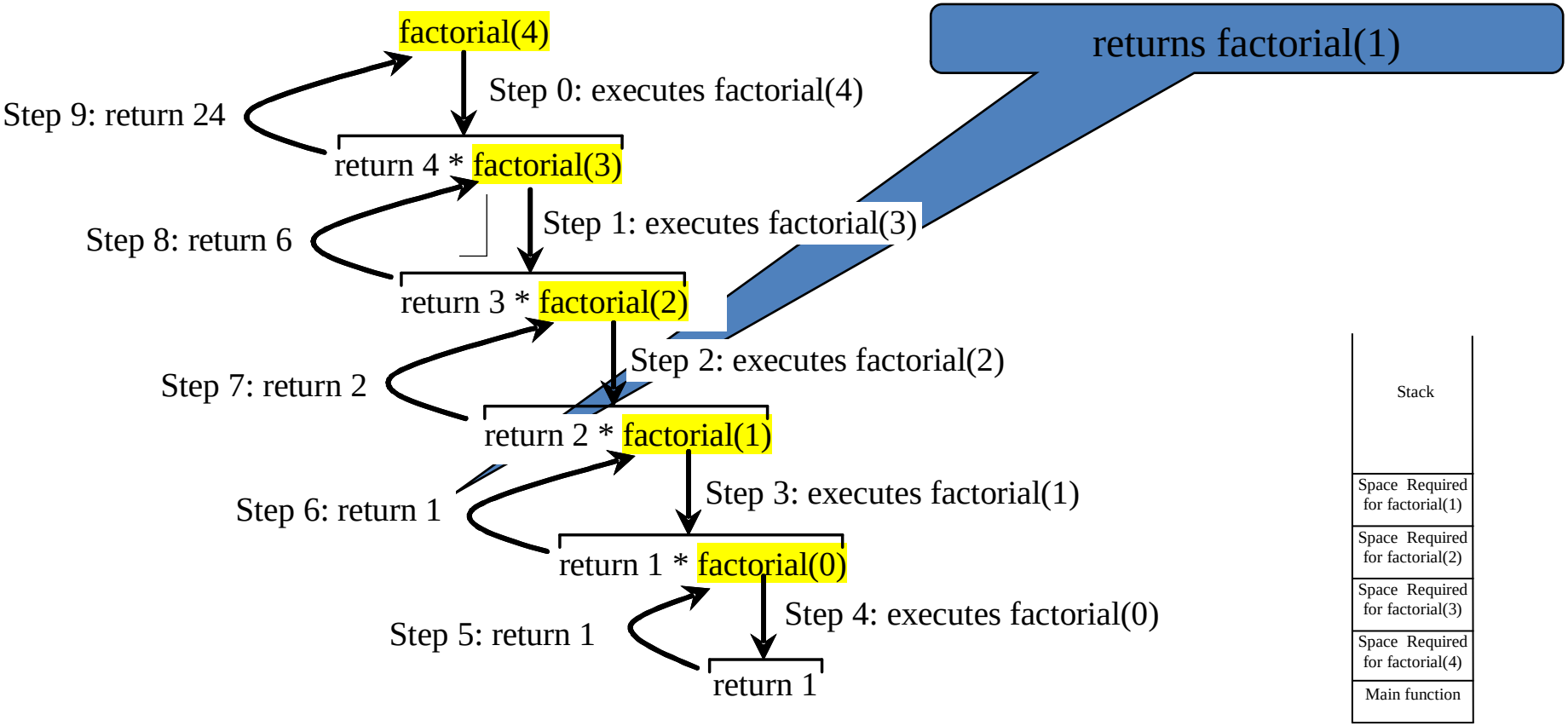
animation

Trace Recursive factorial



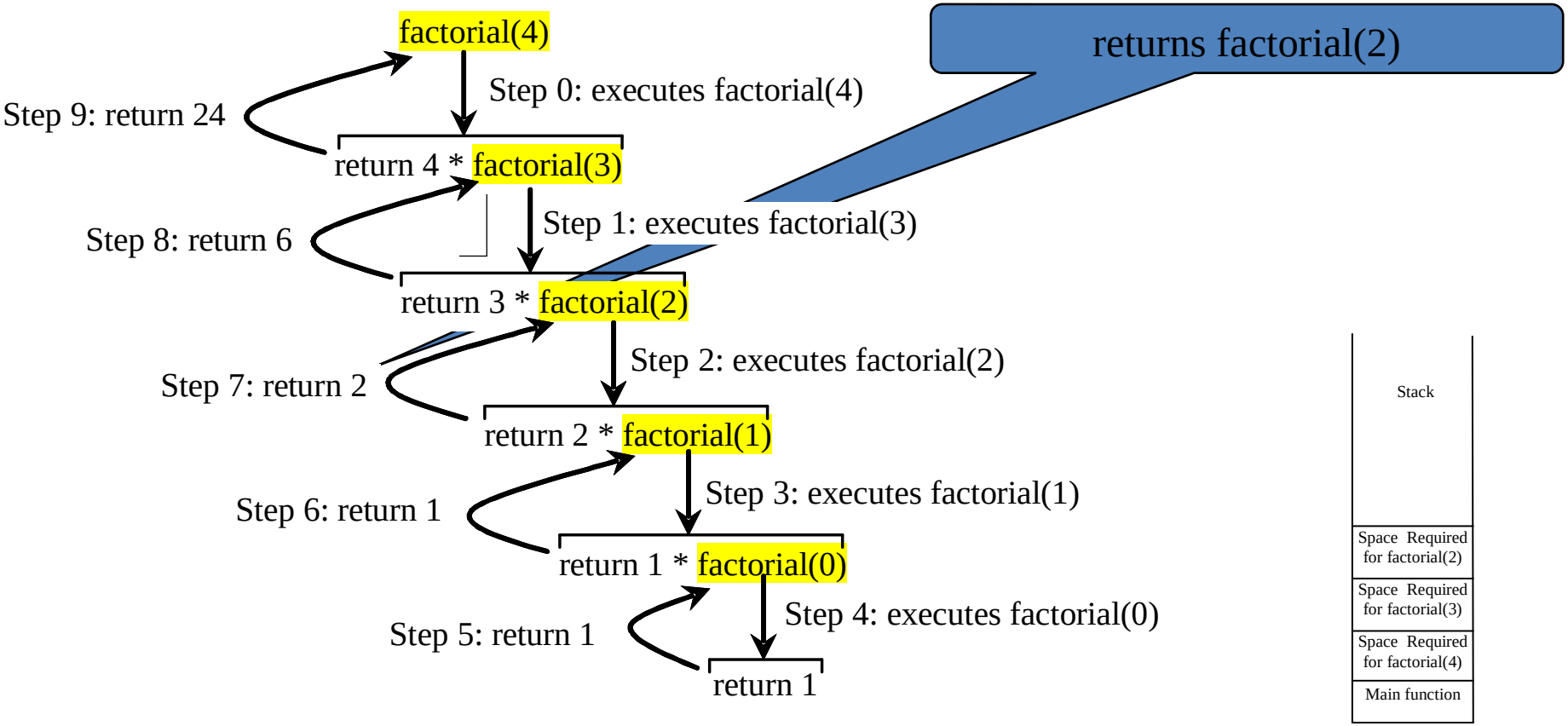
animation

Trace Recursive factorial



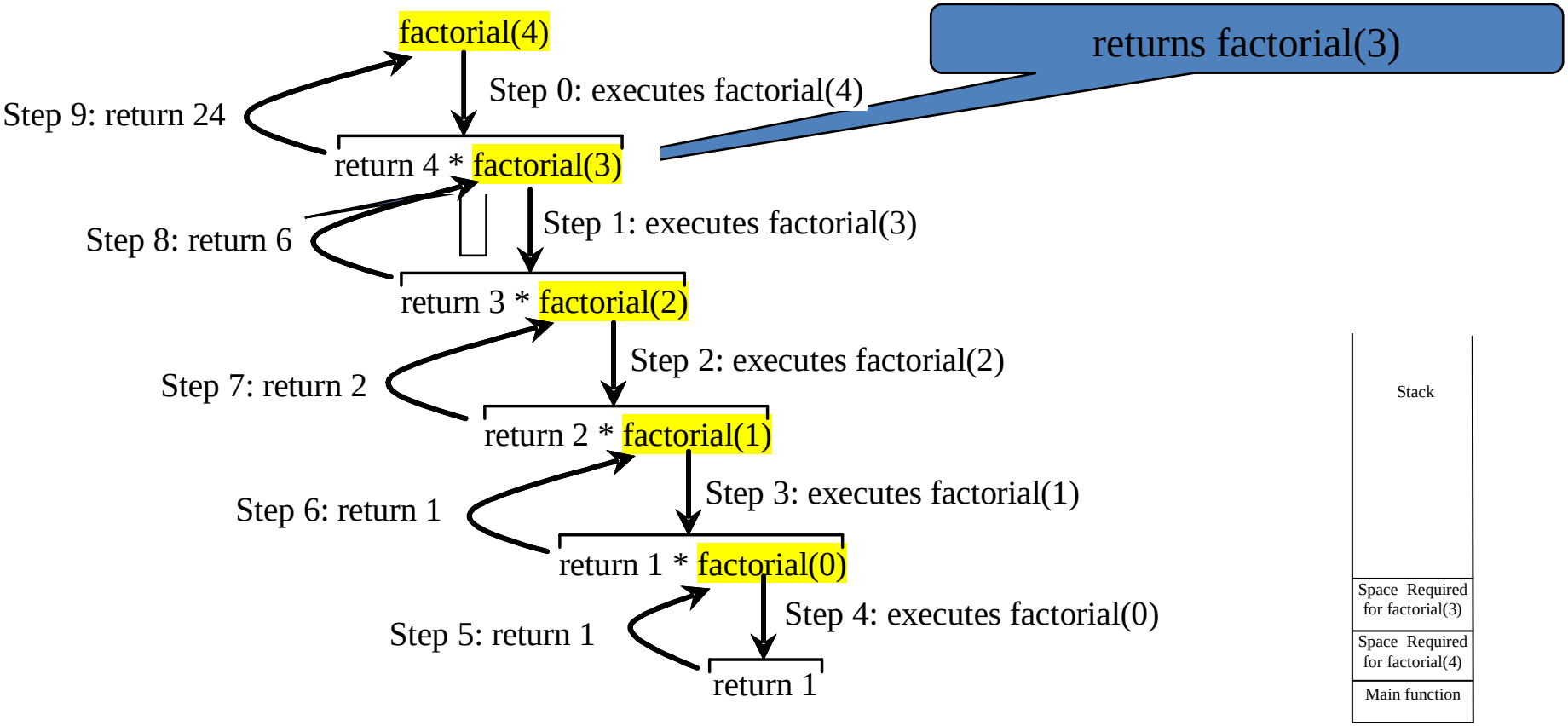
animation

Trace Recursive factorial



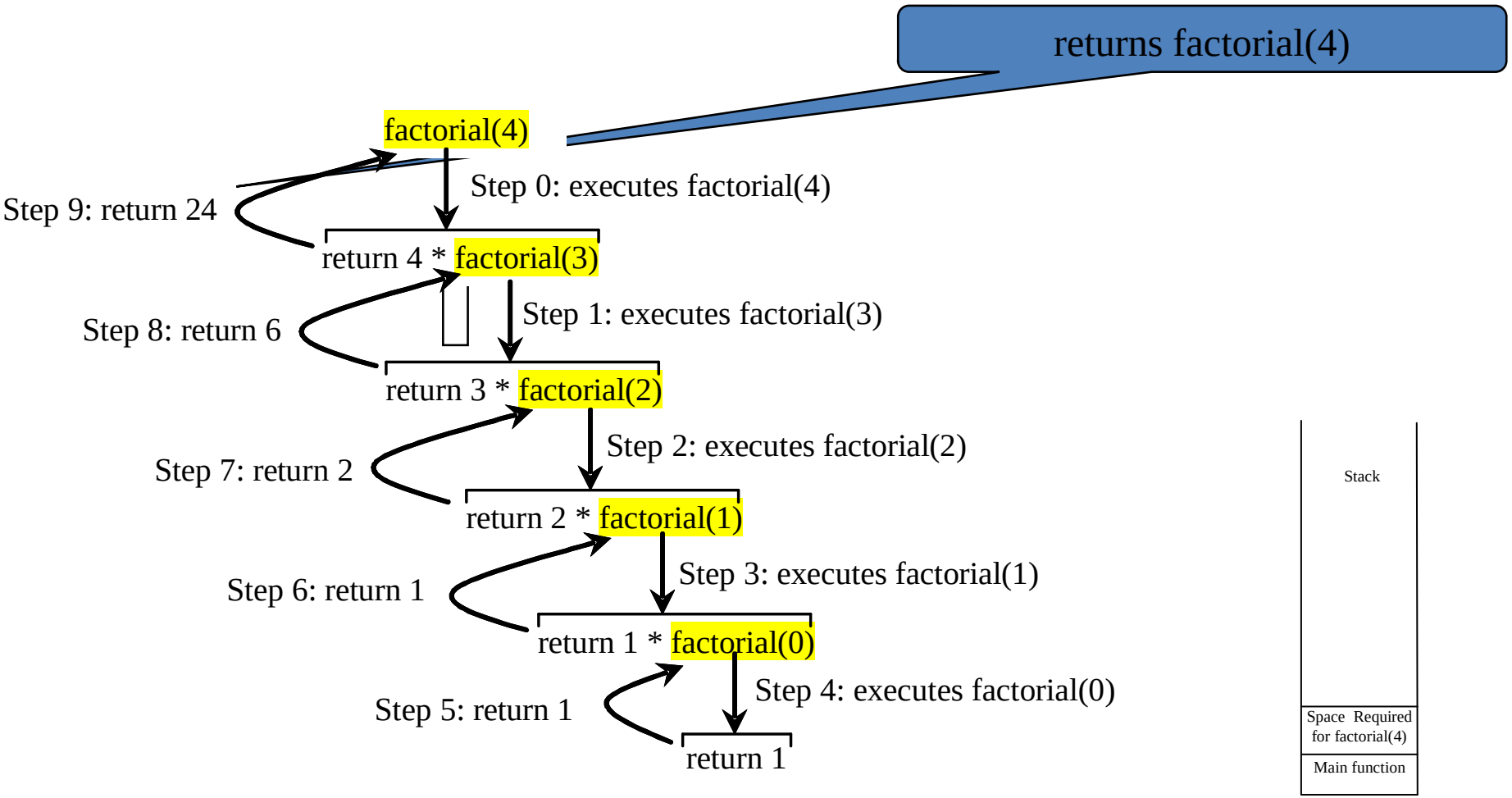
animation

Trace Recursive factorial

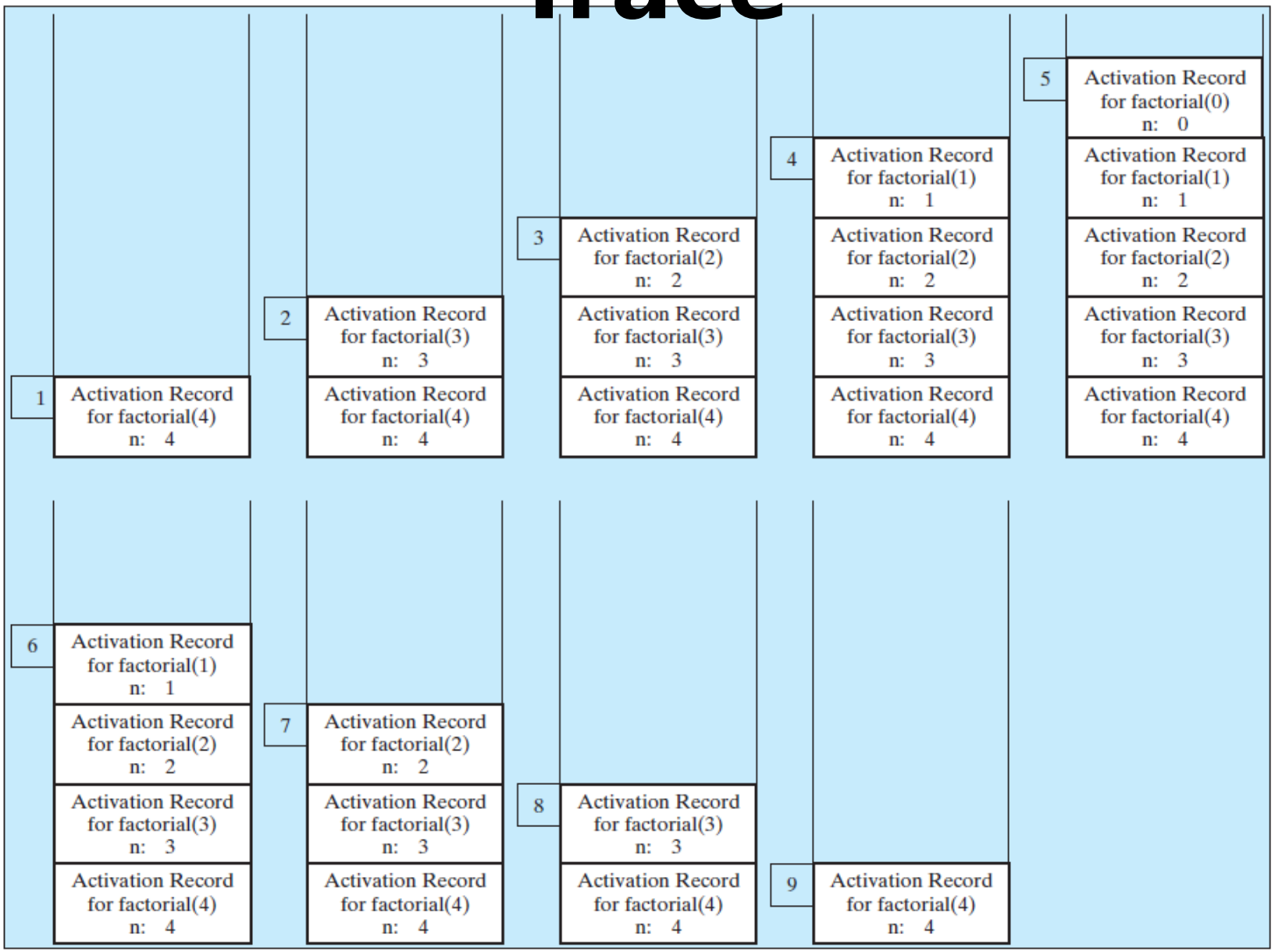


animation

Trace Recursive factorial



factorial(4) Stack Trace



Outline

- Introduction
- Example: Factorials