Recursion

Review of Recursion

- When a function calls itself one or more times (directly or indirectly)
- Form of repetition
- Typically used to perform same operation on a smaller subset and then build the result based on what is returned from the smaller case
- Normally has at least one base case for stopping
- Based on inductive logic

Iteration vs. Recursion

- Anything that can be done iteratively can be do recursively and vice versa
  - Not always a good idea, some problems naturally lend themselves to one mode of thinking or the other

Example: Factorial

- The product of an integer and all that come before it
- **n! = n * (n-1) * (n-2) * … * (n-(n-1)) * 1** (for all n > 0)
- Base Case: **0! = 1**

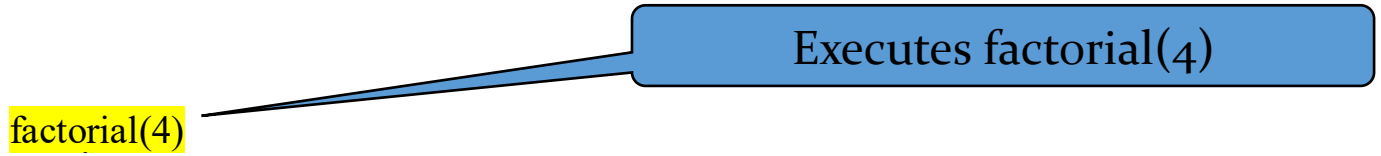
Iterative Factorial

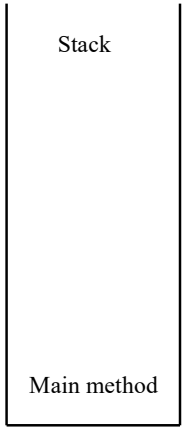
```
int factorial(int n) {
    int fact;
    if (n == 0)
        fact = 1;
    else
        for (fact = n; n > 1; n--)
            fact = fact * (n-1);
    return fact;
}
```

Recursive Factorial

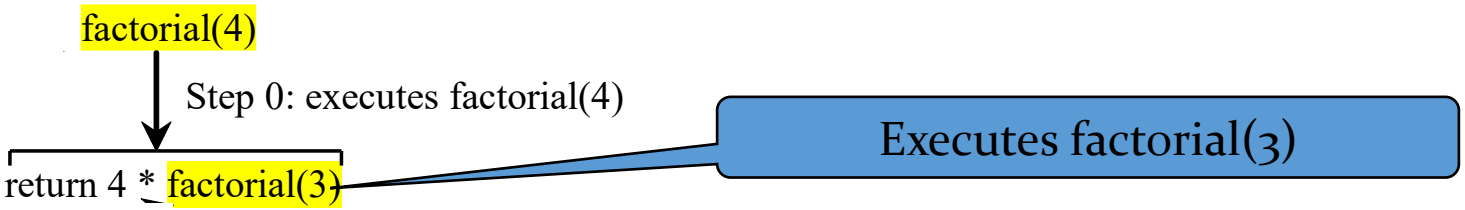
```
int factorial (int n) {
    if (n == 0)
        return 1;
    return n * factorial(n-1);
}
```

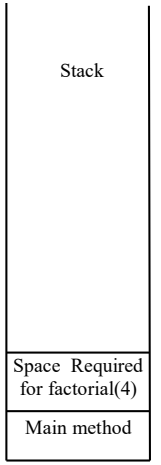
Recursive Factorial



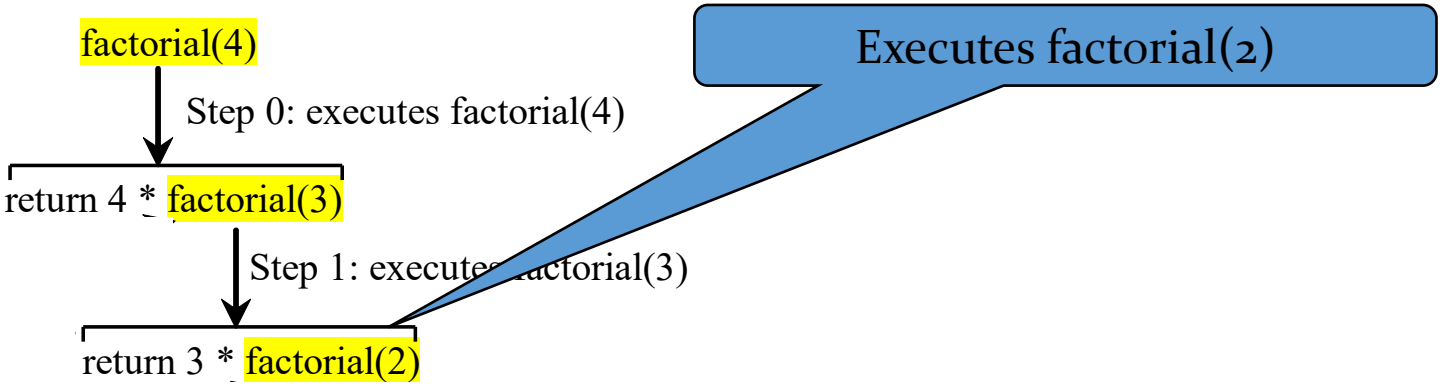


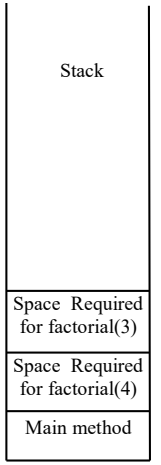
Recursive Factorial



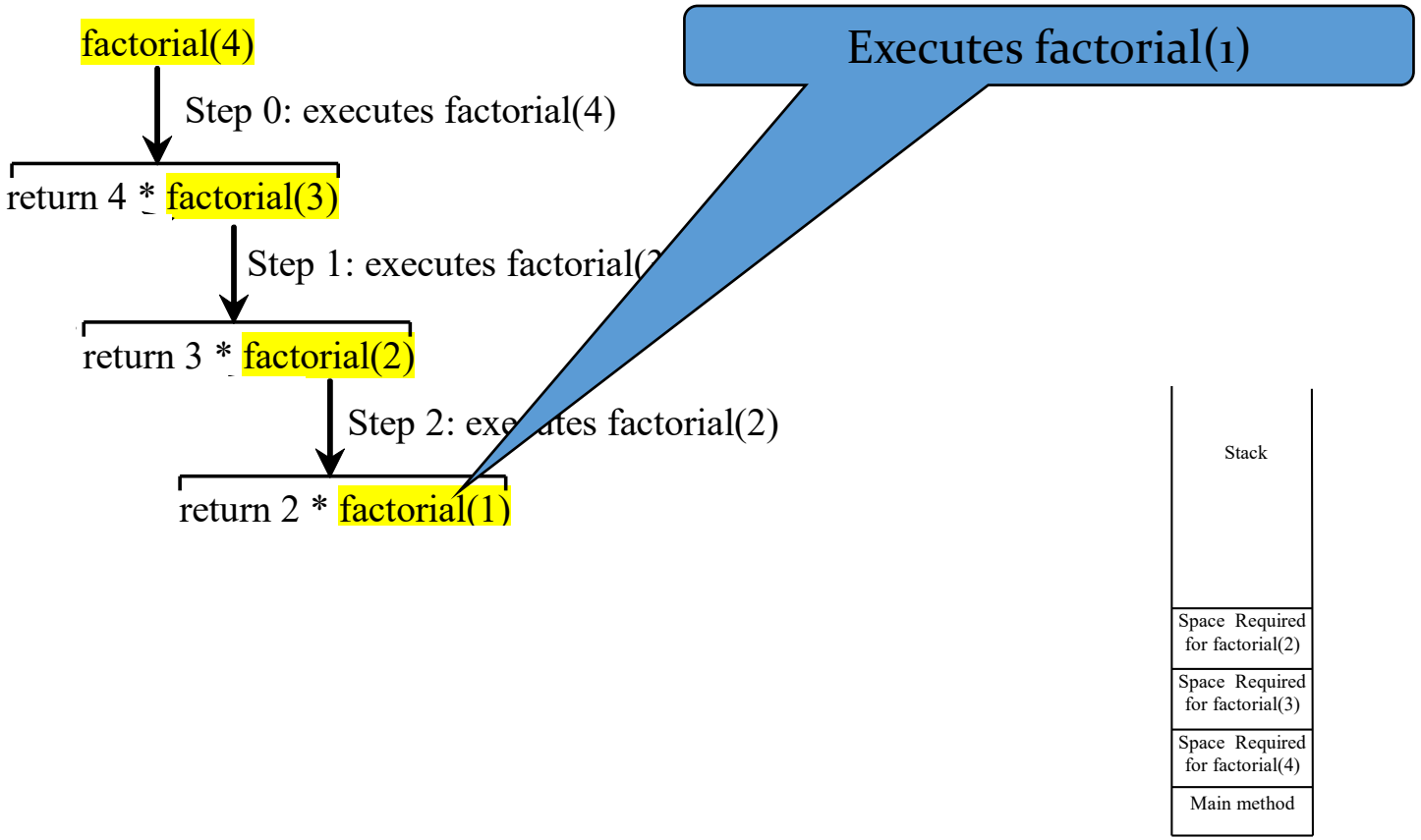


Recursive Factorial

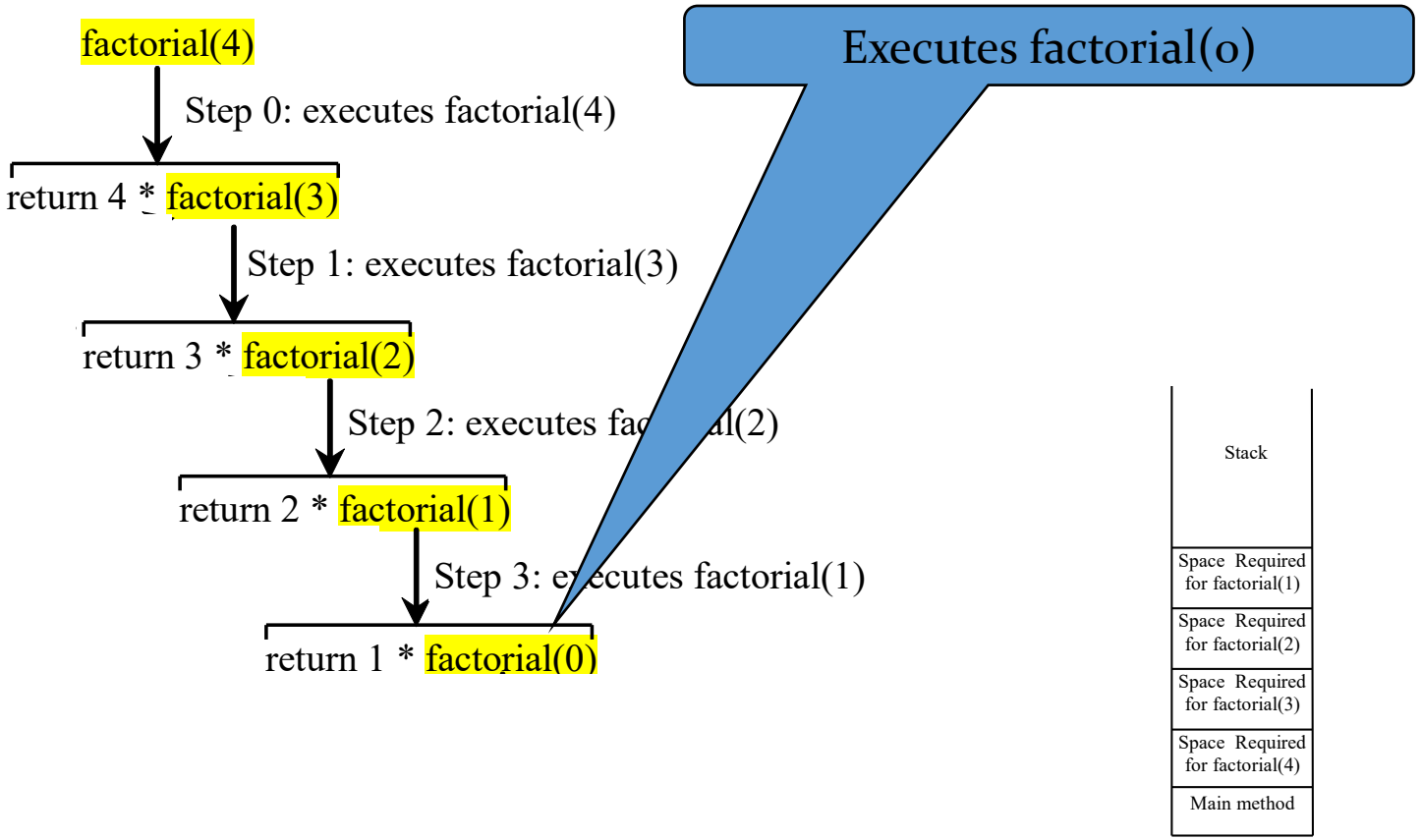




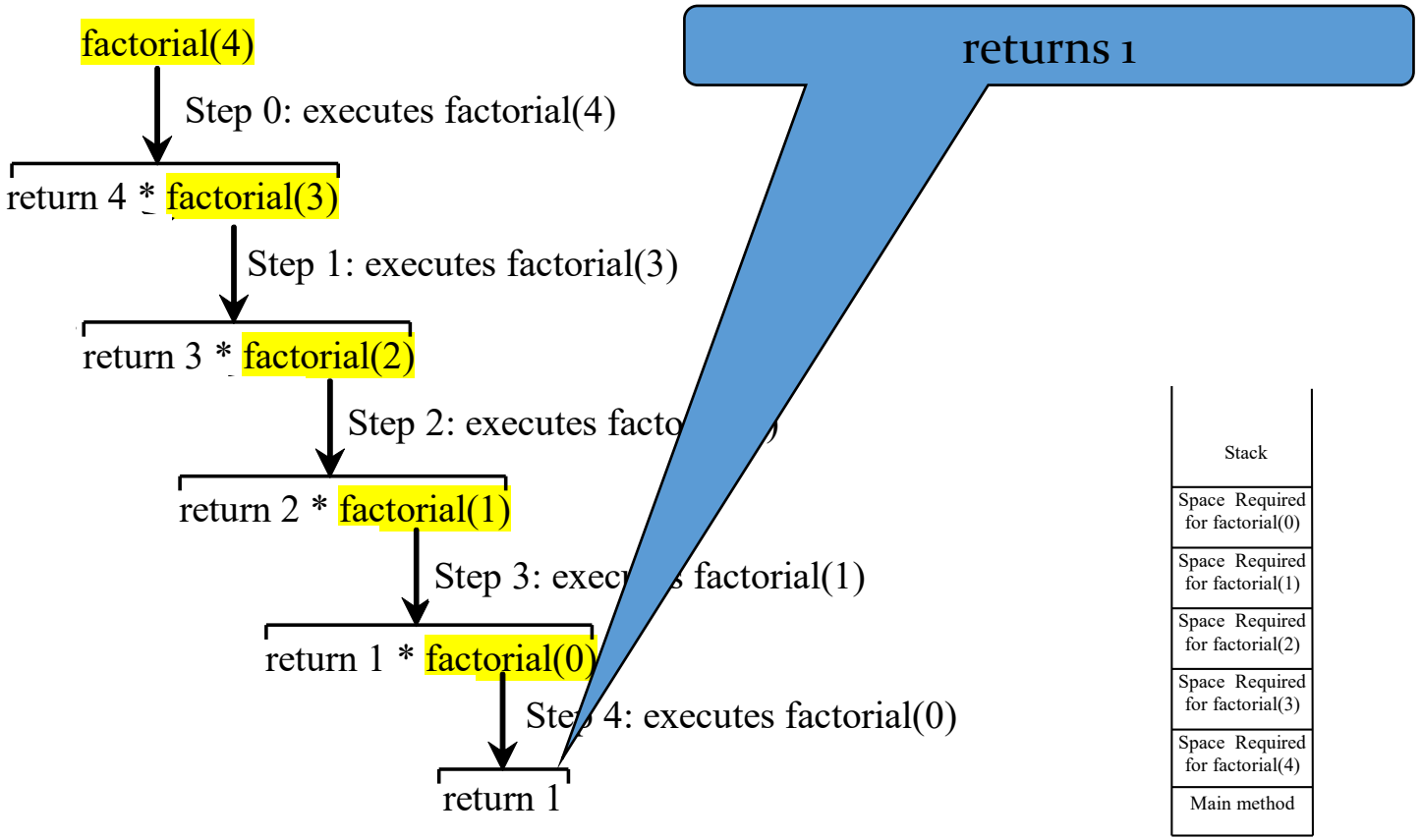
Recursive Factorial



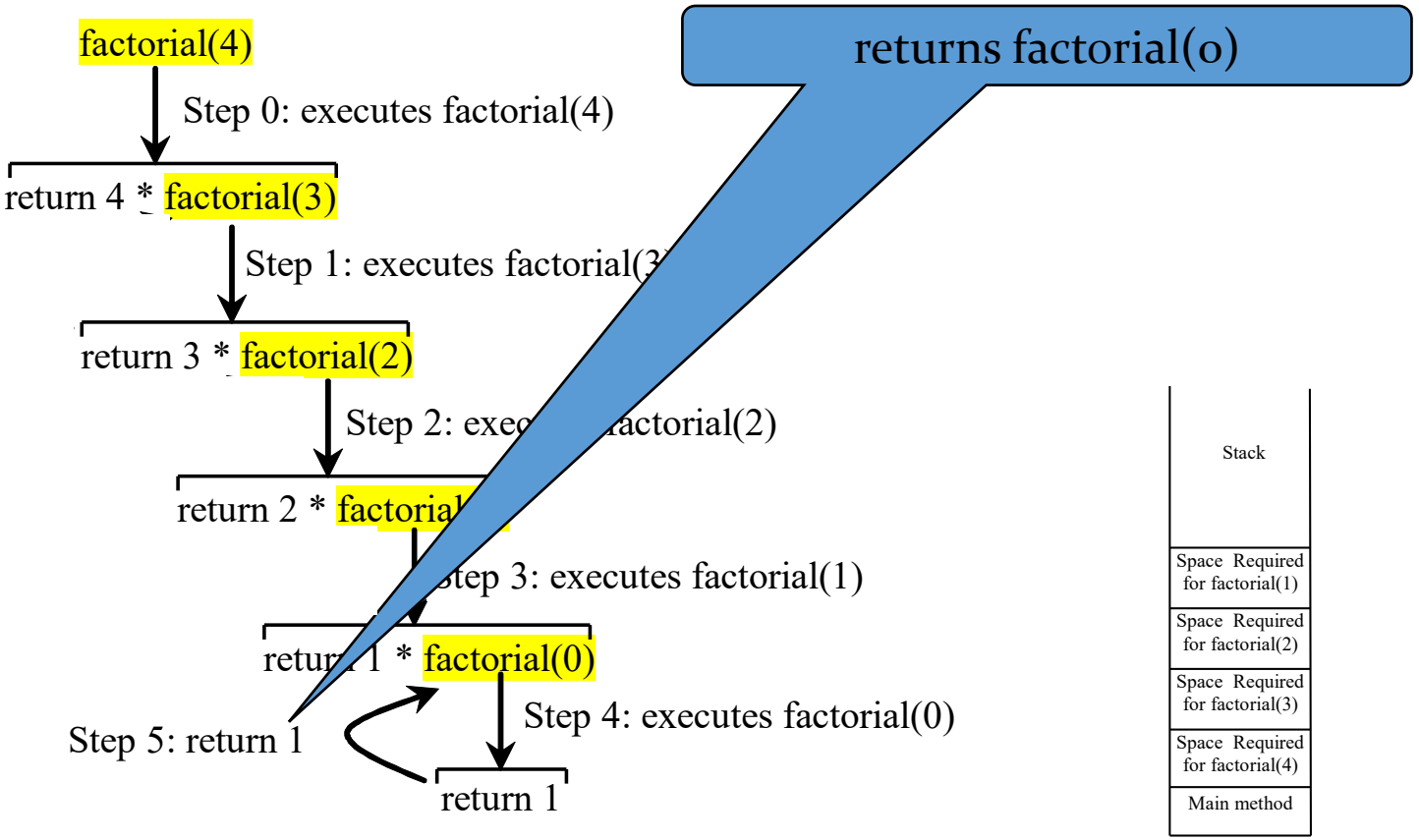
Recursive Factorial



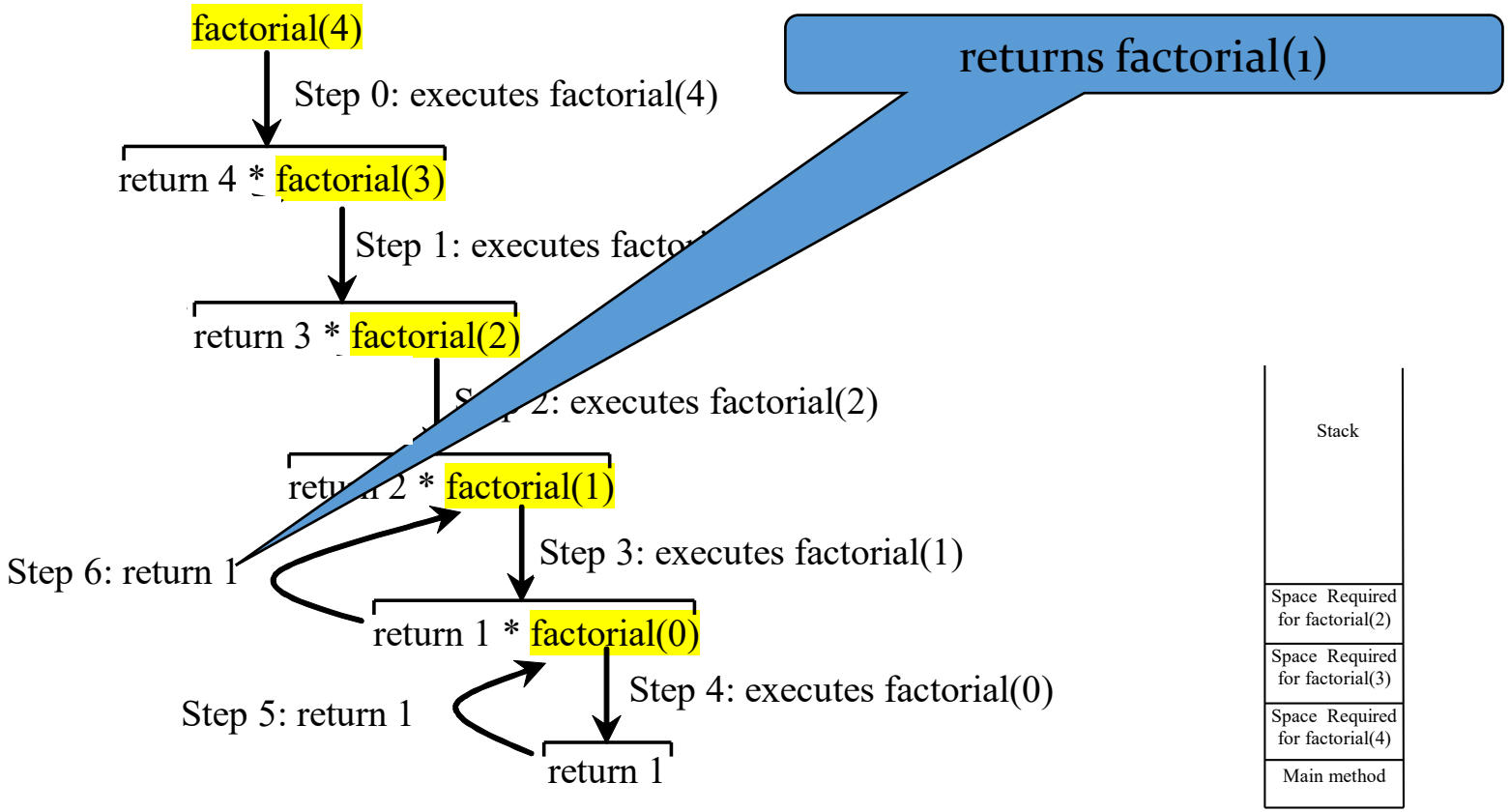
Recursive Factorial



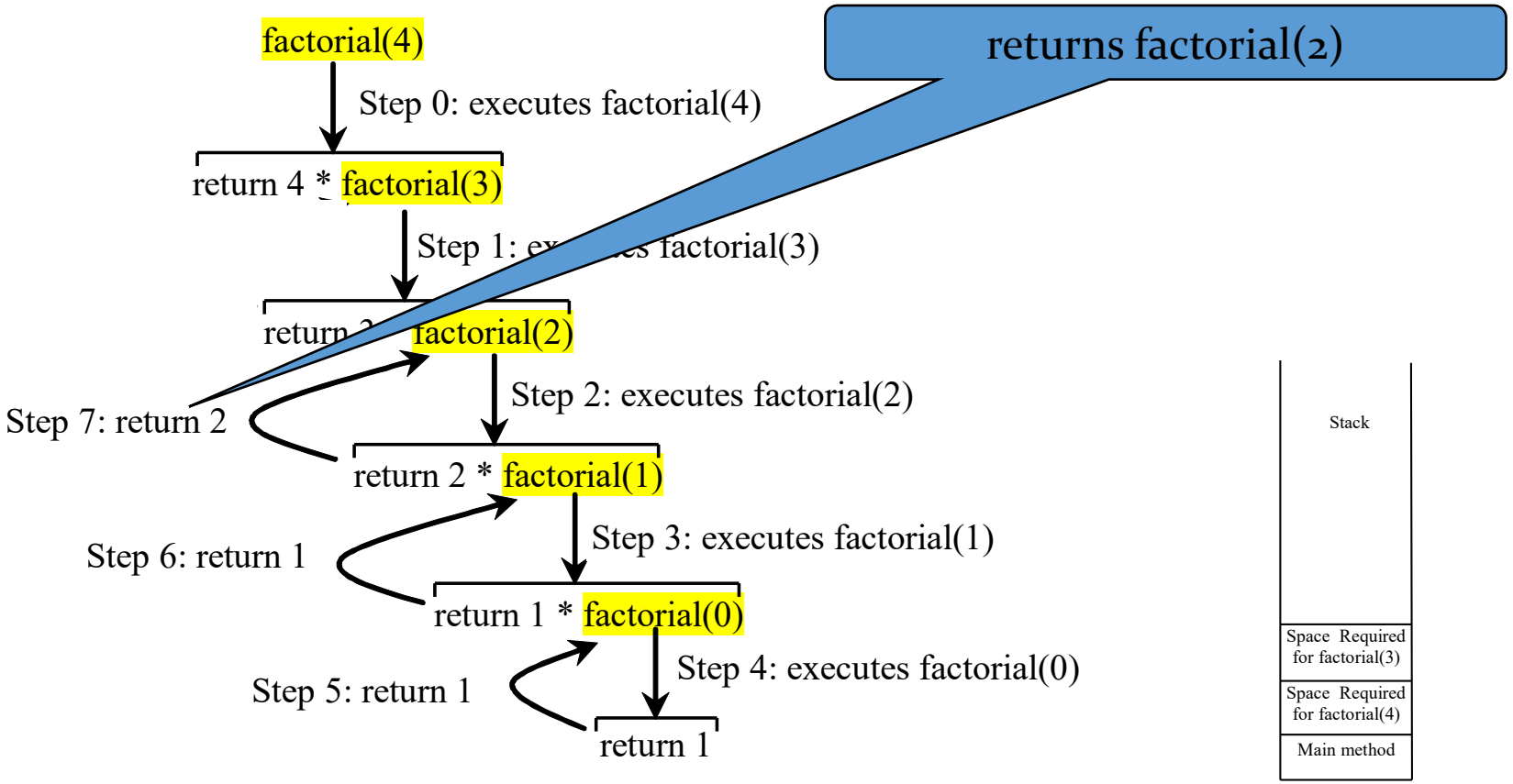
Recursive Factorial



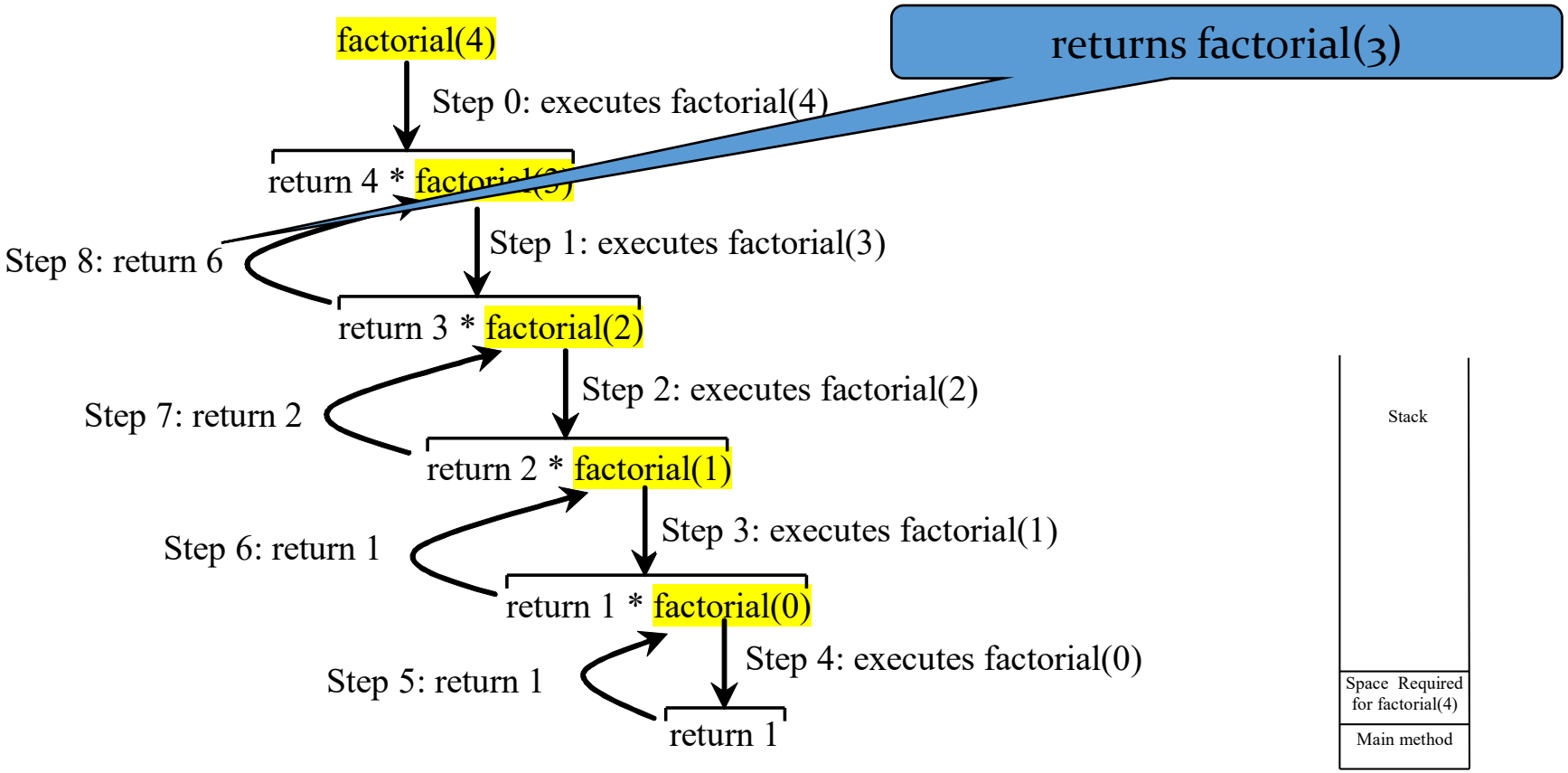
Recursive Factorial



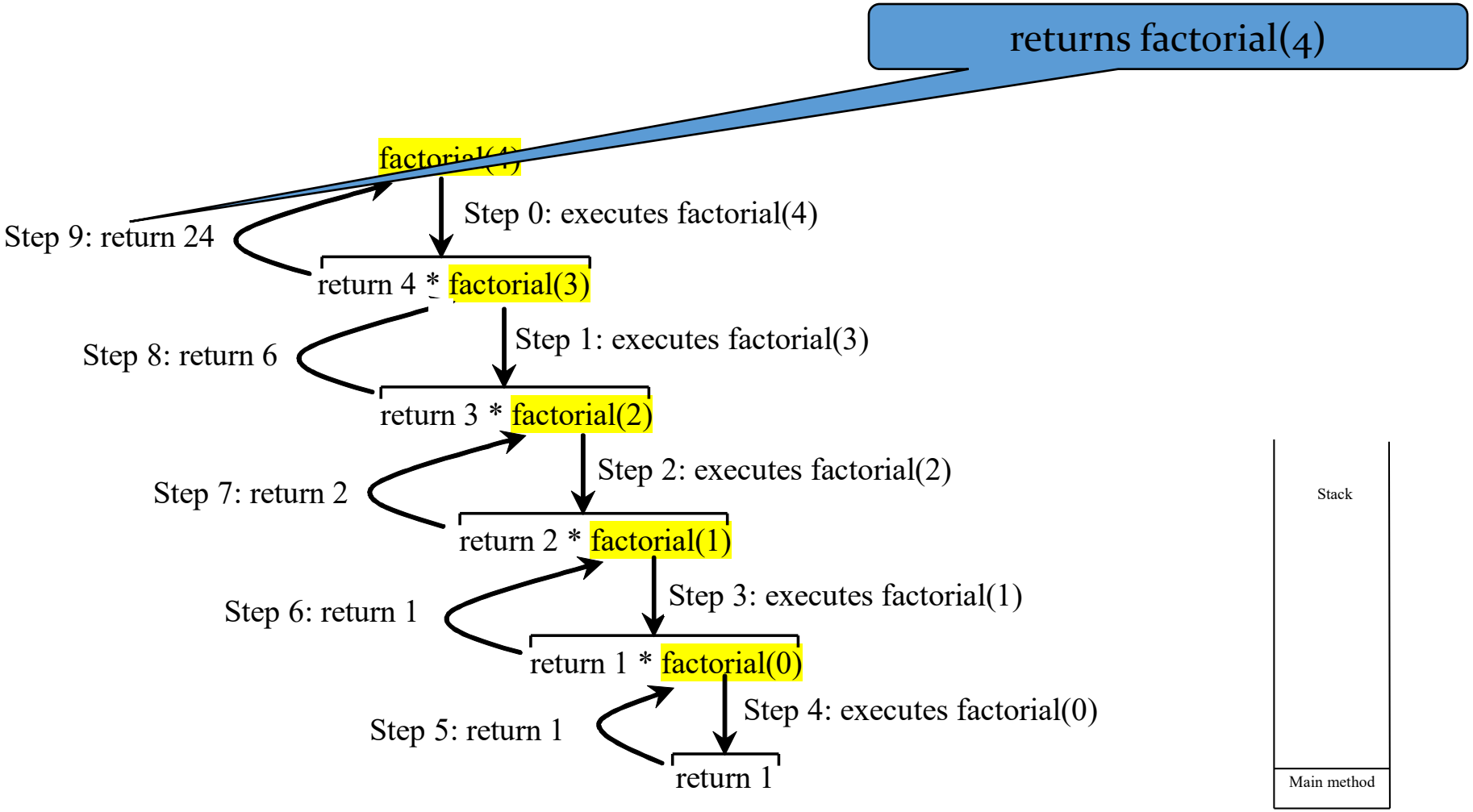
Recursive Factorial



Recursive Factorial



Recursive Factorial



Pros and Cons of Recursion

- Pros
  - Readable
  - Sometimes easier to conceptualize for problems that have many moving parts
- Cons
  - Efficiency
  - Memory usage
    - Each call to the function makes a new function stack frame (as shown on previous slide)

Another Example…

- Investigate Merge Sort
  - https://www.geeksforgeeks.org/merge-sort/

- Can you explain the ordering of the steps?