```java
 5  /** Method that returns the sum of the sequence
 6   * of integers from 1 to n, what about n < 0?
 7   *
 8   * @param n
 9   * @return sum from 1 to n
10   */
11  public int sumOf(int n) {
12      int sum;
13      if (n == 1)
14          sum = 1;
15      else
16          sum = n + sumOf(n-1);
17      return sum;
18  }
```
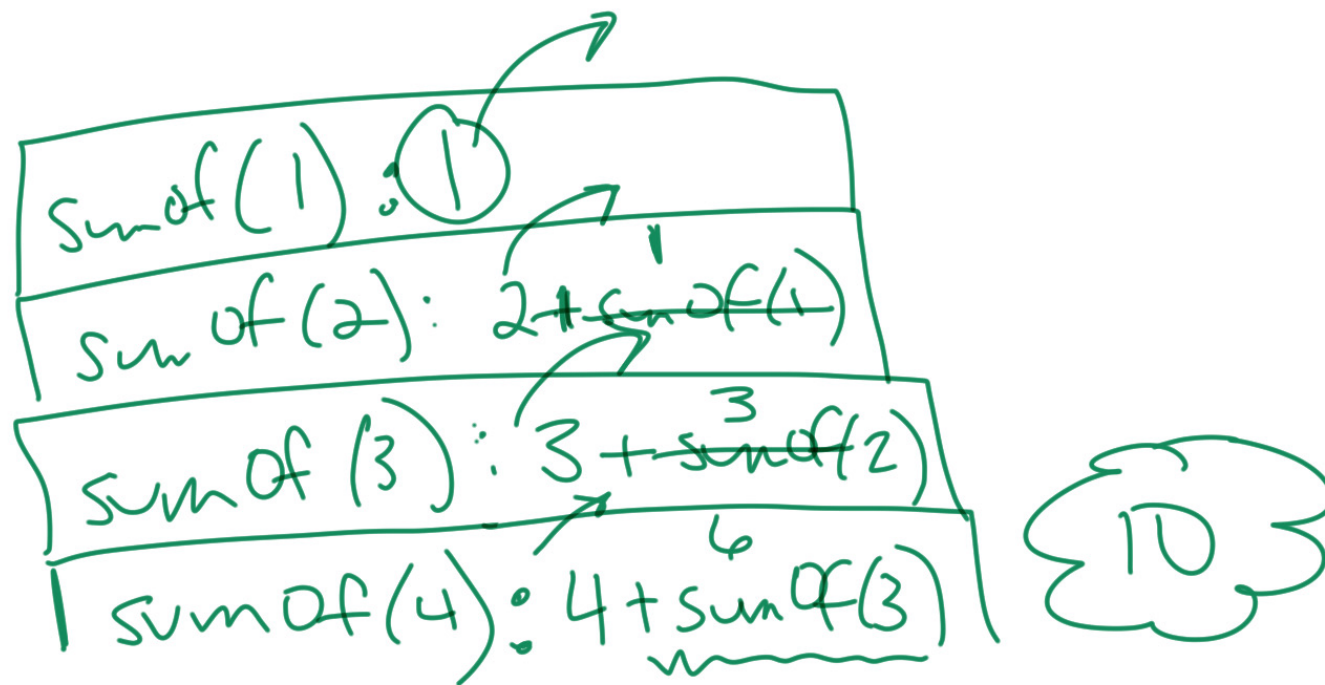
$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \ldots + n$$

Sum of (1) : ①

Sum of (2) : 2 + Sum of (1)

Sum of (3) : 3 + Sum of (2)
$$3$$

Sum of (4) : 4 + Sum of (3)
$$6$$

10

```java
21   /**
22    * Compute computes N!  =  1 * 2 *  ...  * N
23    * @param n
24    * @return n!
25    */
26   public int factorialRecurse(int n) {
27       if (n > 1)
28           return (n * factorialRecurse(n-1));
29       else
30           return 1;
31   }
32
```

Recursive Execution Trace

Factorial (5)

return 120

5 * Factorial (4)

return 24

4 * Factorial (3)

return 6

3 * Factorial (2)

return 2

2 * Factorial (1)

1

return 1

First the "recursive descent" . . .

. . . and then the return sequence

Virginia Tech 2020