Lecture 10

Recursion

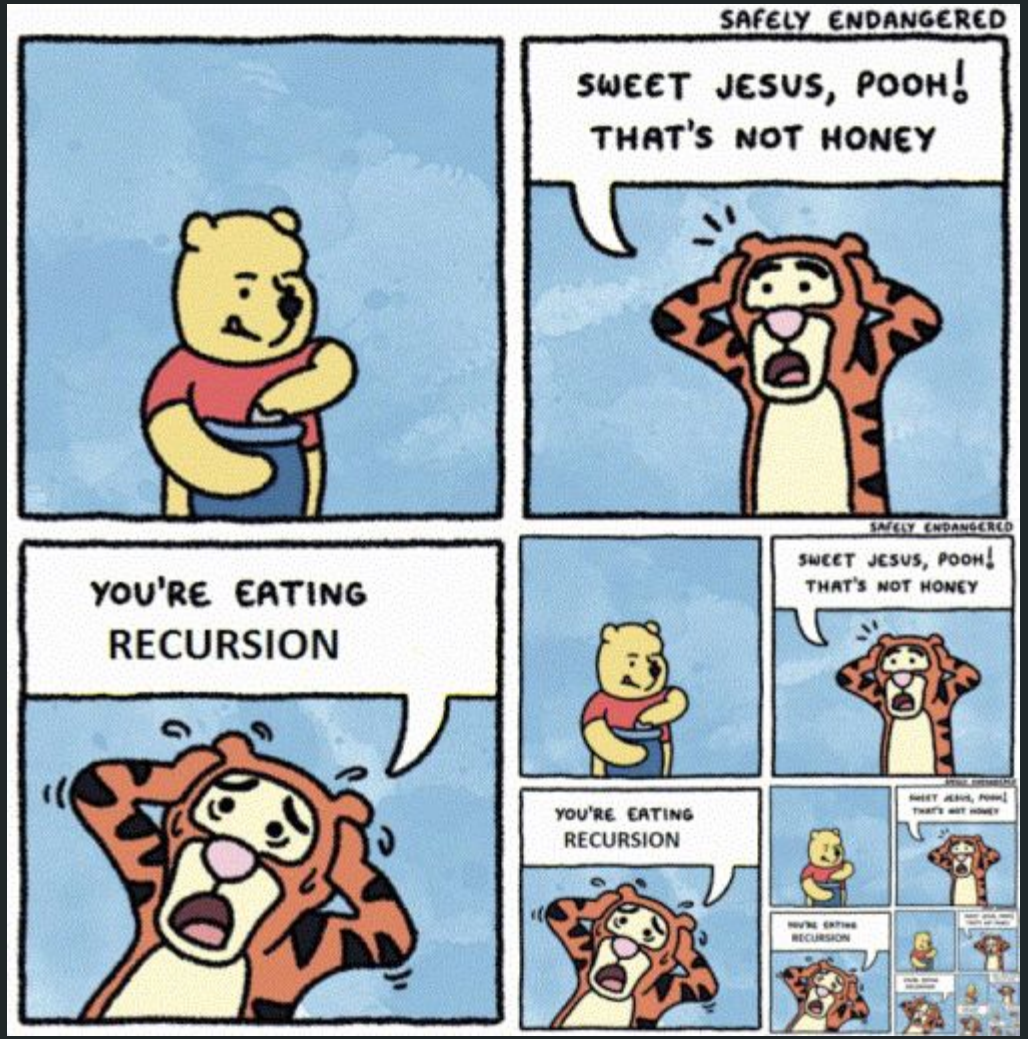
What is recursion?

Recursion, by definition, is “when a thing is defined in terms of itself.”

A recursive function is a function that calls itself, either directly, or indirectly (through another function).

What is recursion?

Example:



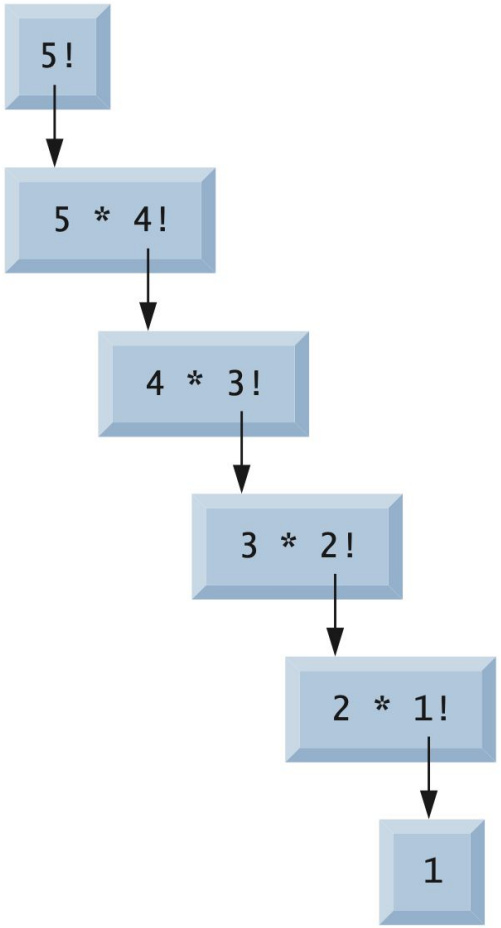
Key components of recursion

- Base case
  - condition to which the recursion converges
  - when it is reached, the recursion stops
- Problem reduction
  - with each recursive function call the problem should become smaller and be closer to the base case

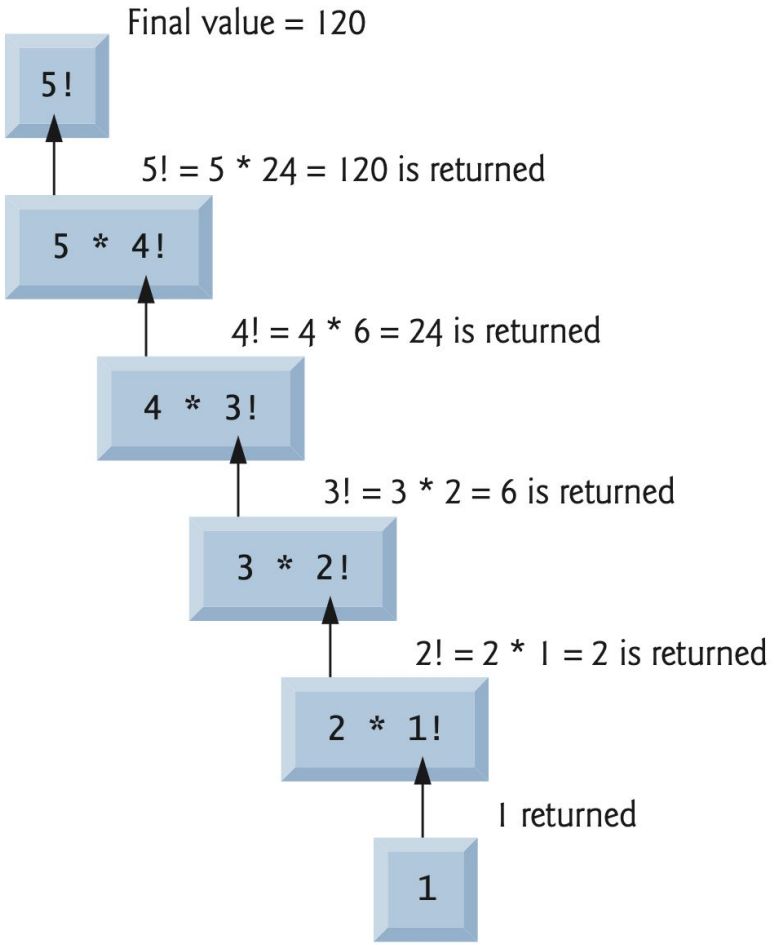
Factorial

$$n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 1$$

$$n! = n \cdot (n-1)!$$



(a) Procession of recursive calls

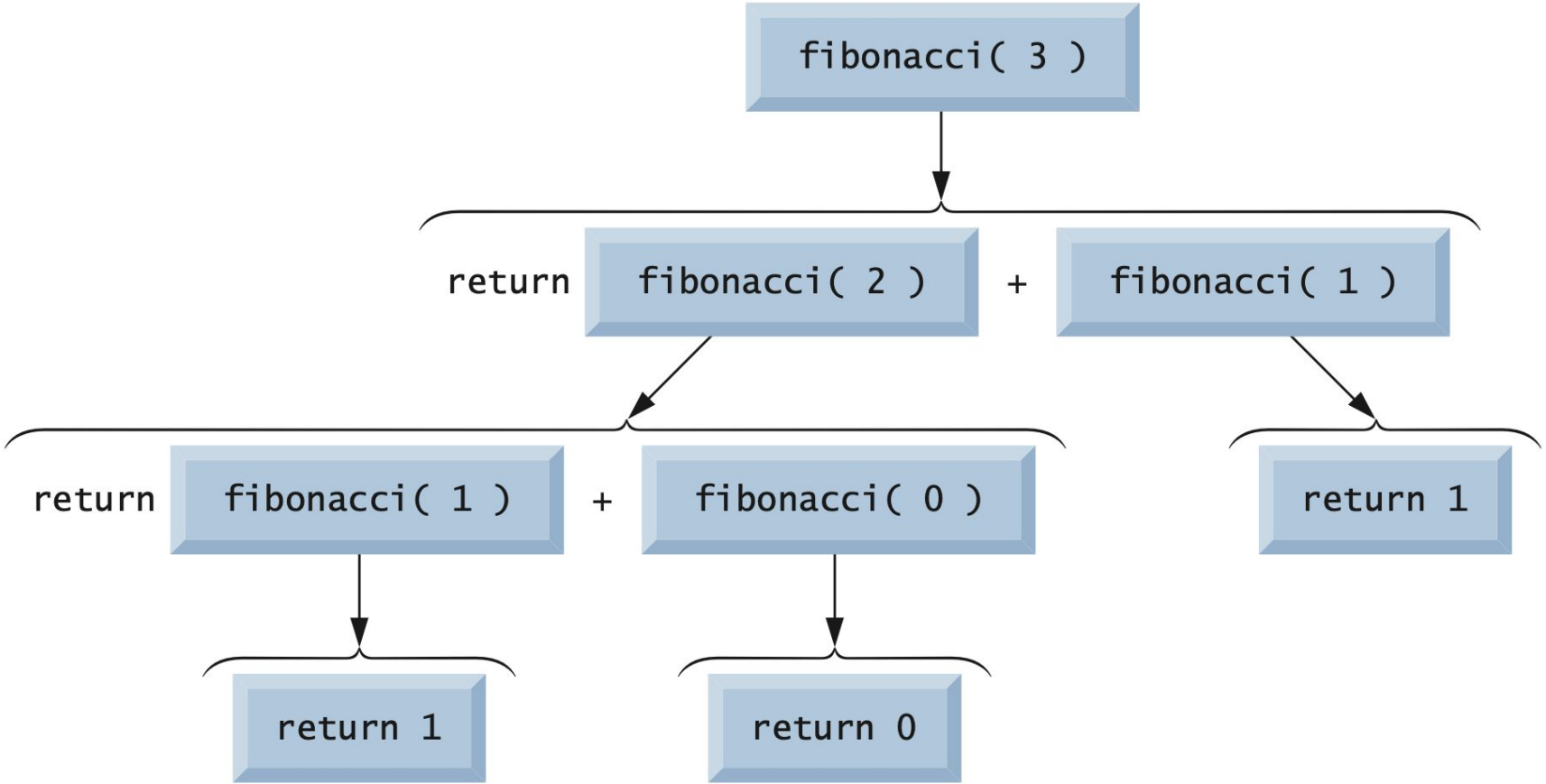


(b) Values returned from each recursive call

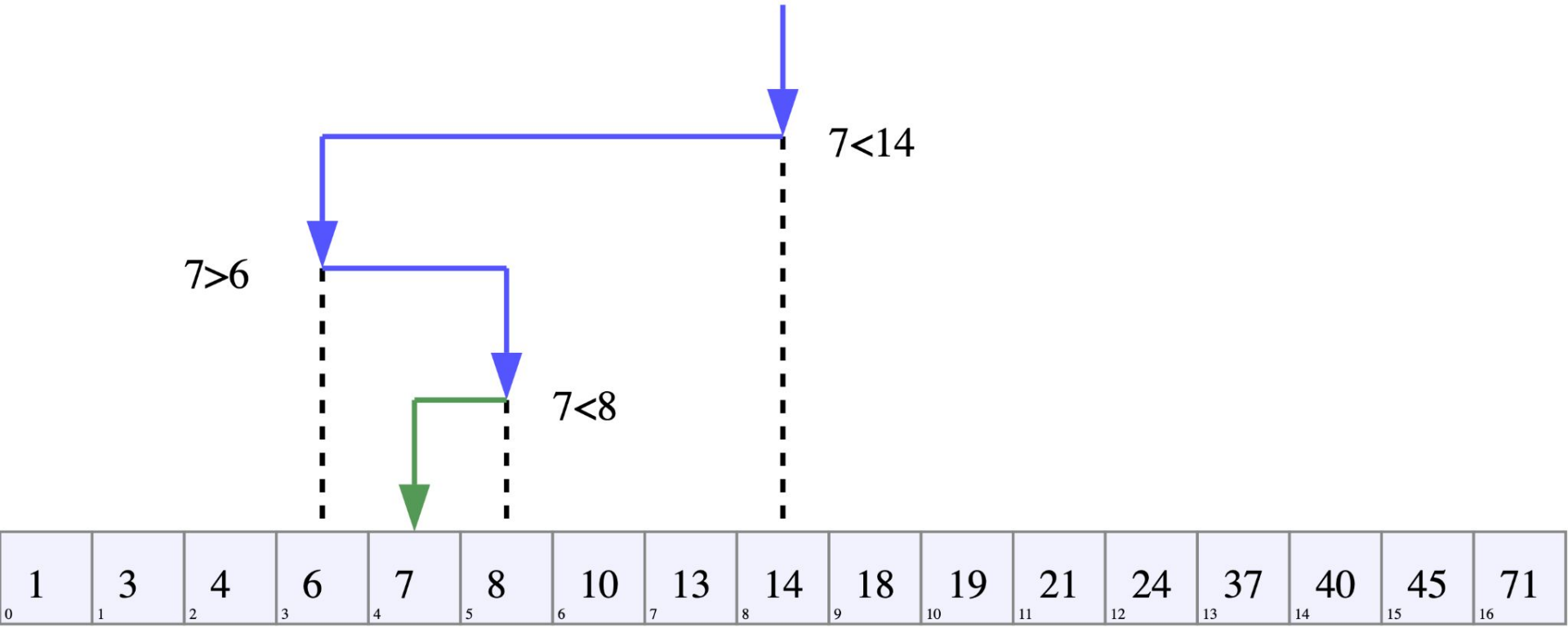
Fibonacci series

$$0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

$$\text{fibonacci}(0) = 0$$
$$\text{fibonacci}(1) = 1$$
$$\text{fibonacci}(n) = \text{fibonacci}(n-1) + \text{fibonacci}(n-2)$$


fibonacci(3)
return
fibonacci(2)
+
fibonacci(1)
return
fibonacci(1)
+
fibonacci(0)
return 1
return 1
return 0

Binary search algorithm



Binary search algorithm

- Only works with sorted arrays
- Takes up to log2(n) comparisons to find the target, where n is the number of elements in the array

Recursion vs. Iteration

Both iteration and recursion are based on a control statement: Iteration uses a repetition structure; recursion uses a selection structure. Both iteration and recursion involve repetition: Iteration explicitly uses a repetition structure; recursion achieves repetition through repeated function calls. Iteration and recursion both involve a termination test: Iteration terminates when the loop-continuation condition fails; recursion terminates when a base case is recognized. Iteration with counter-controlled repetition and recursion both gradually approach termination: Iteration modifies a counter until the counter assumes a value that makes the loop-continuation condition fail; recursion produces simpler versions of the original problem until the base case is reached. Both iteration and recursion can occur infinitely: An infinite loop occurs with iteration if the loop-continuation test never becomes false;infinite recursion occurs if the recursion step does not reduce the problem during each recursive call in a manner that converges on the base case.

Recursion vs. Iteration

- Any problem that can be solved recursively can also be solved iteratively (non-recursively).
- A recursive approach is normally chosen *when the recursive approach more naturally mirrors the problem* and *results in a program that's easier to understand and debug*.
- Another reason to choose a recursive solution is that an iterative solution is not apparent.

Additional materials

- Paper:
  - C++ How to Program, Seventh Edition, H. M. Deitel, P. J. Deitel:
    - Chapter 6, Sections 6.19 - 6.21 (available in the KBTU library);
- Digital:
  - informatics.msk.ru:
    - Функции: Условия задач
    - Функции и процедуры. Рекурсия
  - w3schools:
    - C++ Function Recursion