

6. RECURSION

[Hengfeng Wei \(魏恒峰\)](#)

hfw@nju.edu.cn



Nov. 10, 2023

Review

Functions

Function Definition

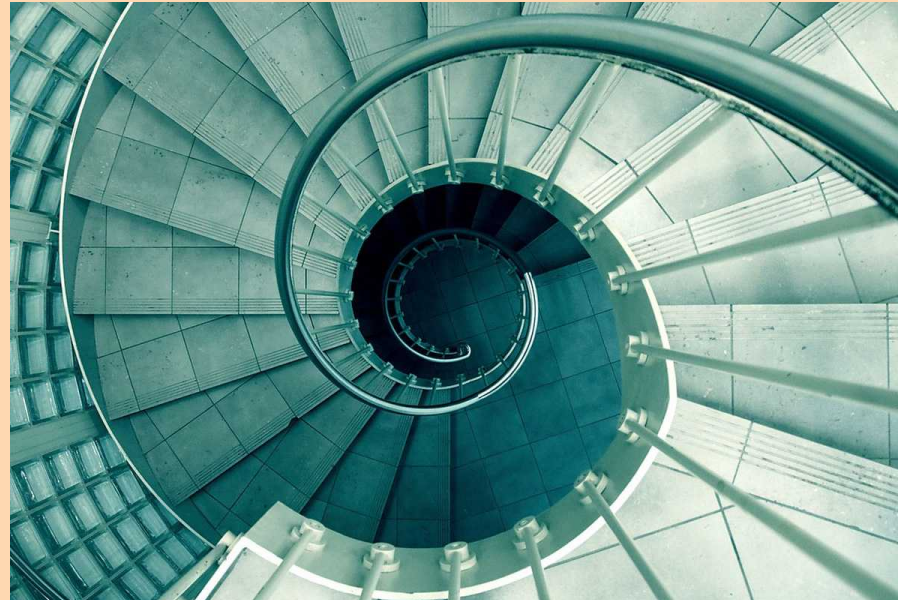
Function Declaration

Arrays as Parameters

Pass by Value

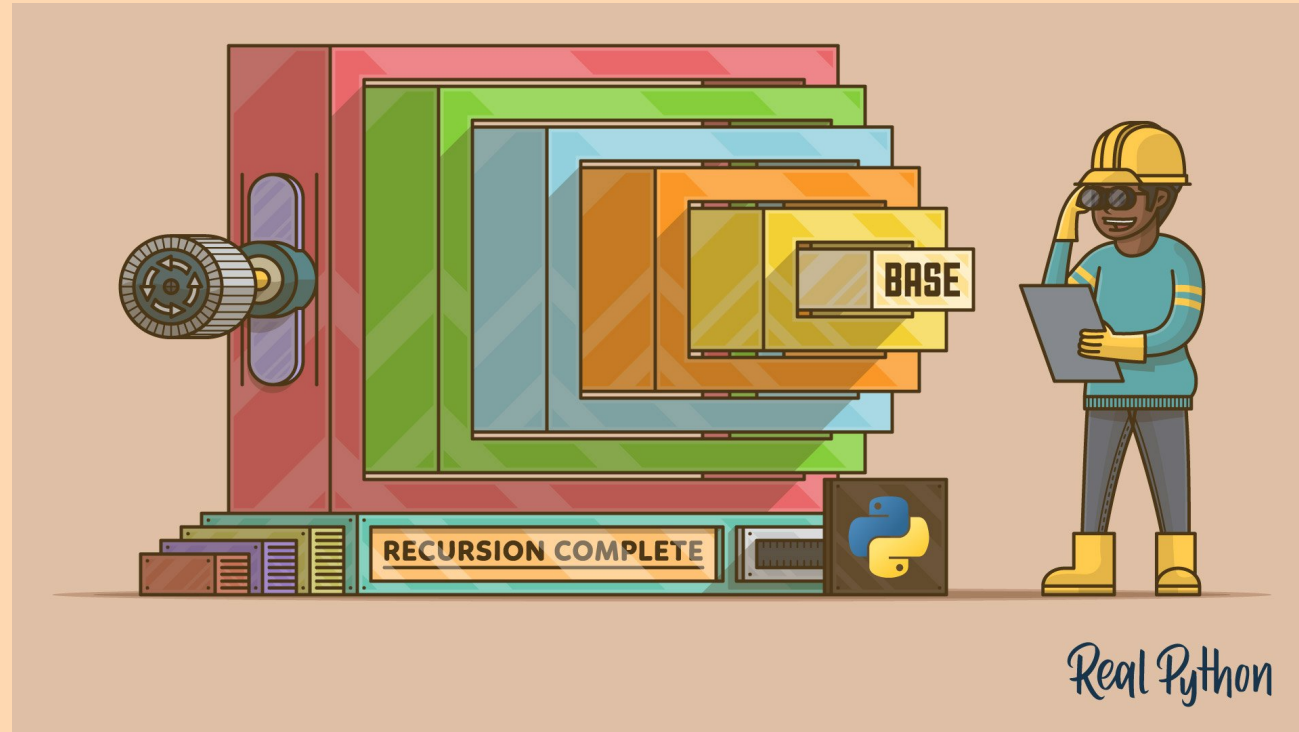
Overview

Recursive Functions (Recursion)



A function that calls itself (**main-re.c**).

(1) Thinking like a Computer Scientist



Solving a task by first solving its smaller subtasks

(1) Thinking like a Computer Scientist

You want to solve a task and suppose you have the **Mirror**.



The **Mirror can solve the smaller task for you magically.**

(1) Thinking like a Computer Scientist

- What is a smaller task? (★★★★★)
- How to solve the task given the solution to the smaller one? (★★★)
- What is the smallest task? (★)

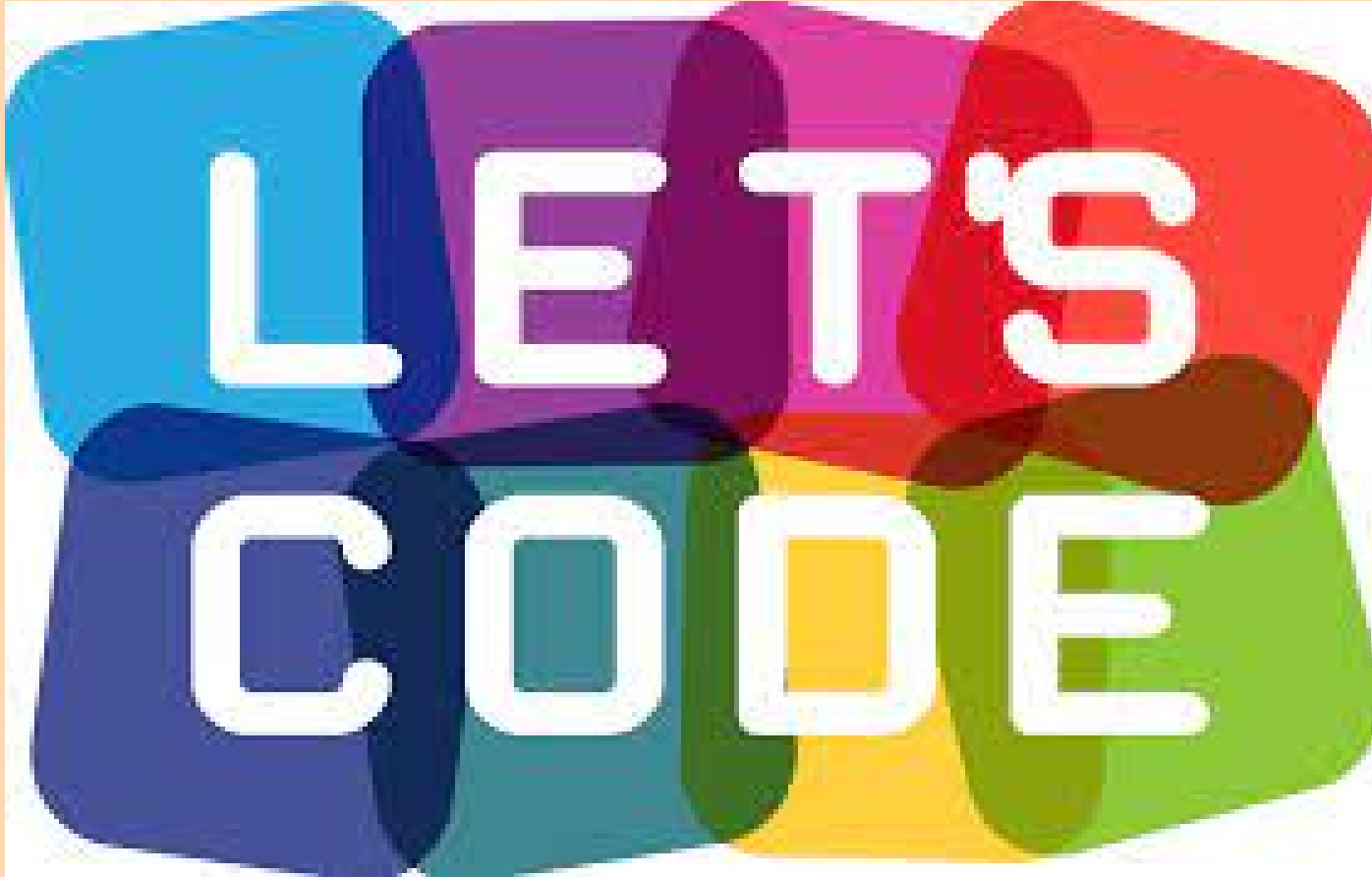
Thinking Recursively

It will be a **loooooooooong** way to go to
master **RECURSION!!!**

(2) Thinking like a Computer



min.c



min-re.c sum-re.c fib-re.c gcd-re.c
bsearch-re.c mergesort.c

Min (**min-re.c**)



Min (**min-re.c**)

$$\begin{aligned}\text{Min}(3, 5, 2, 7) &= \min(7, \text{Min}(3, 5, 2)) \\ &= \min(7, \min(2, \text{Min}(3, 5))) \\ &= \min(7, \min(2, \min(5, \text{Min}(3)))) \\ &= \min(7, \min(2, \min(5, 3))) \\ &= \min(7, \min(2, 3)) \\ &= \min(7, 2) \\ &= 2\end{aligned}$$

Sum ([sum-re.c](#))



Sum (**sum-re.c**)

$$\begin{aligned}\text{Sum}(1, 3, 5, 7) &= 7 + \text{Sum}(1, 3, 5) \\ &= 7 + (5 + \text{Sum}(1, 3)) \\ &= 7 + (5 + (3 + \text{Sum}(1))) \\ &= 7 + (5 + (3 + 1)) \\ &= 7 + (5 + 4) \\ &= 7 + 9 \\ &= 16\end{aligned}$$

Fibonacci Sequence (**fib-re.c**)

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad (n > 1)$$

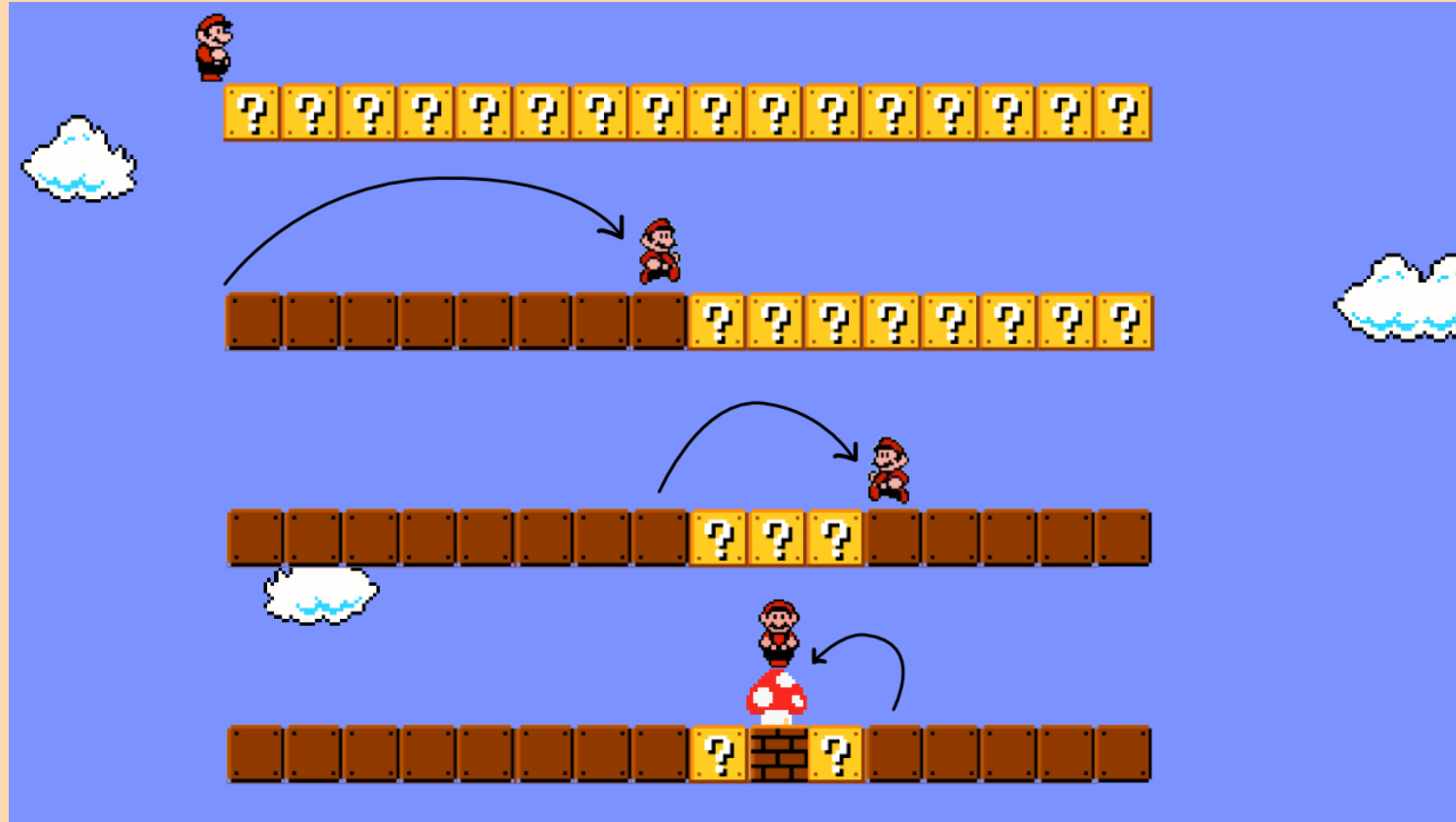


Greatest Common Divisor (**gcd-re.c**)

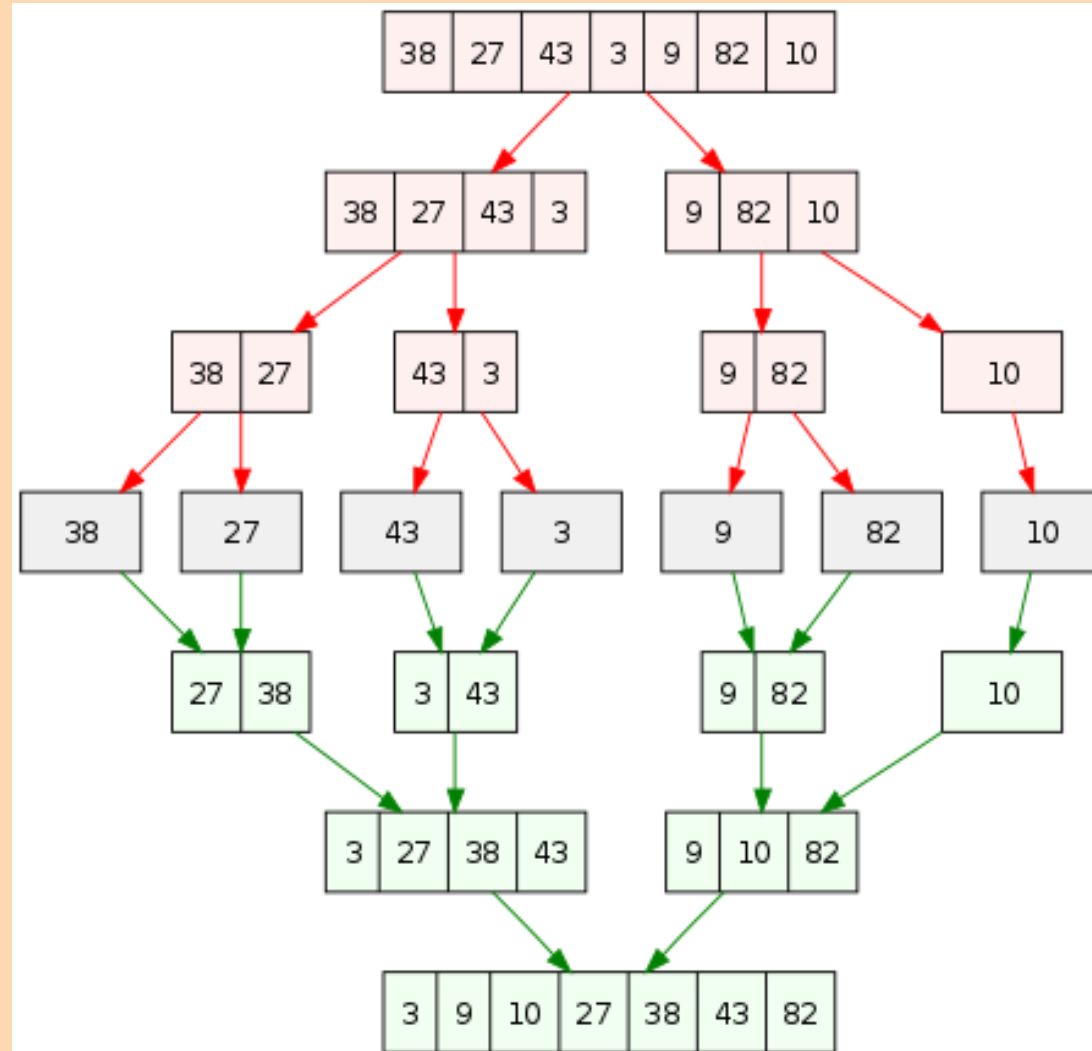


$$\text{gcd}(a, b) = \text{gcd}(b, a \% b)$$

Binary Search (**bsearch-re.c**)



MergeSort (mergesort.c)



MergeSort (**mergesort.c**)

6 5 3 1 8 7 2 4

