

# 13. RECURSION

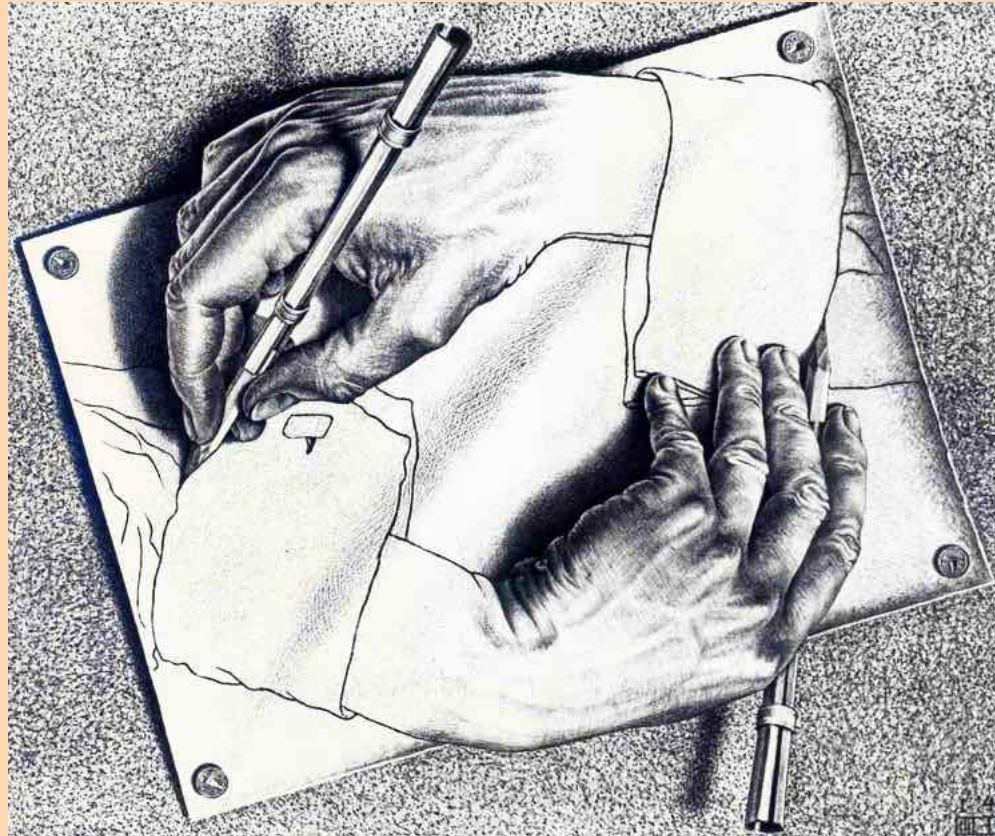
[Hengfeng Wei \(魏恒峰\)](#)

[hfwei@nju.edu.cn](mailto:hfwei@nju.edu.cn)



Dec. 27, 2024

# Recursive Functions (Recursion)



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

# (1) Thinking like a Computer Scientist

You have a problem and suppose you have the **Mirror**.



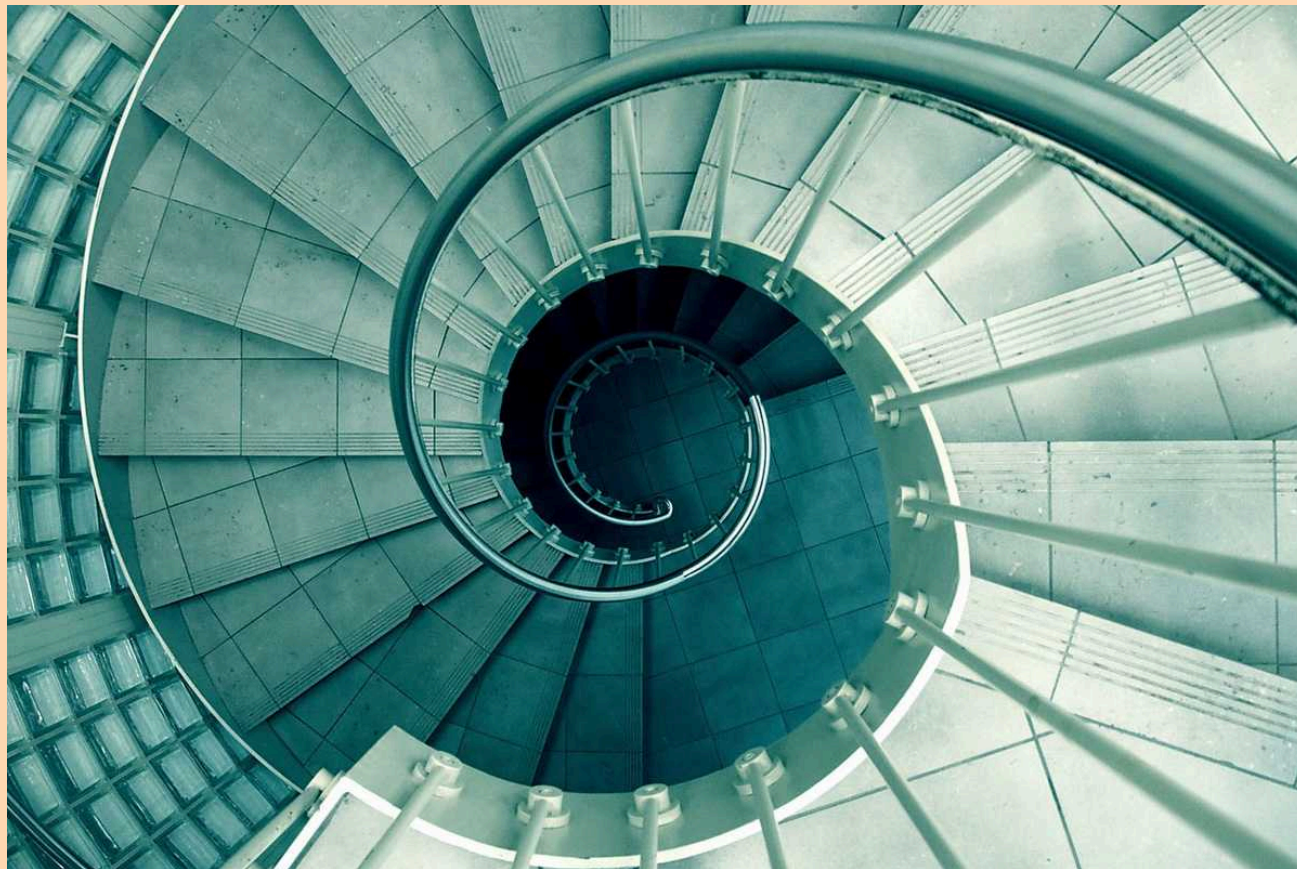
**The **Mirror** can solve smaller sub-problems for you magically.**

# (1) Thinking like a Computer Scientist

- How to reduce the original problem into smaller sub-problems?
- How to solve the original problem given the solutions to the smaller sub-problems?

**What are the smaller sub-problems? (★ ★ ★ ★ ★)**





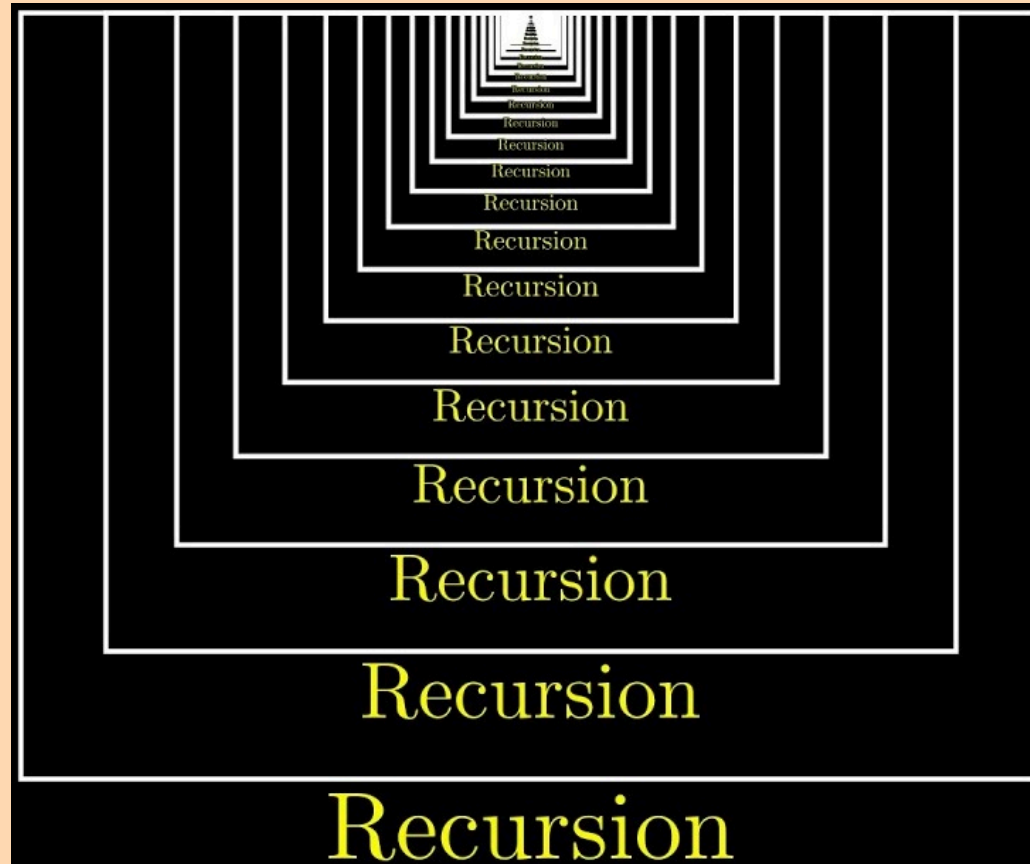
**stairs.c**

## (2) Thinking like a Computer

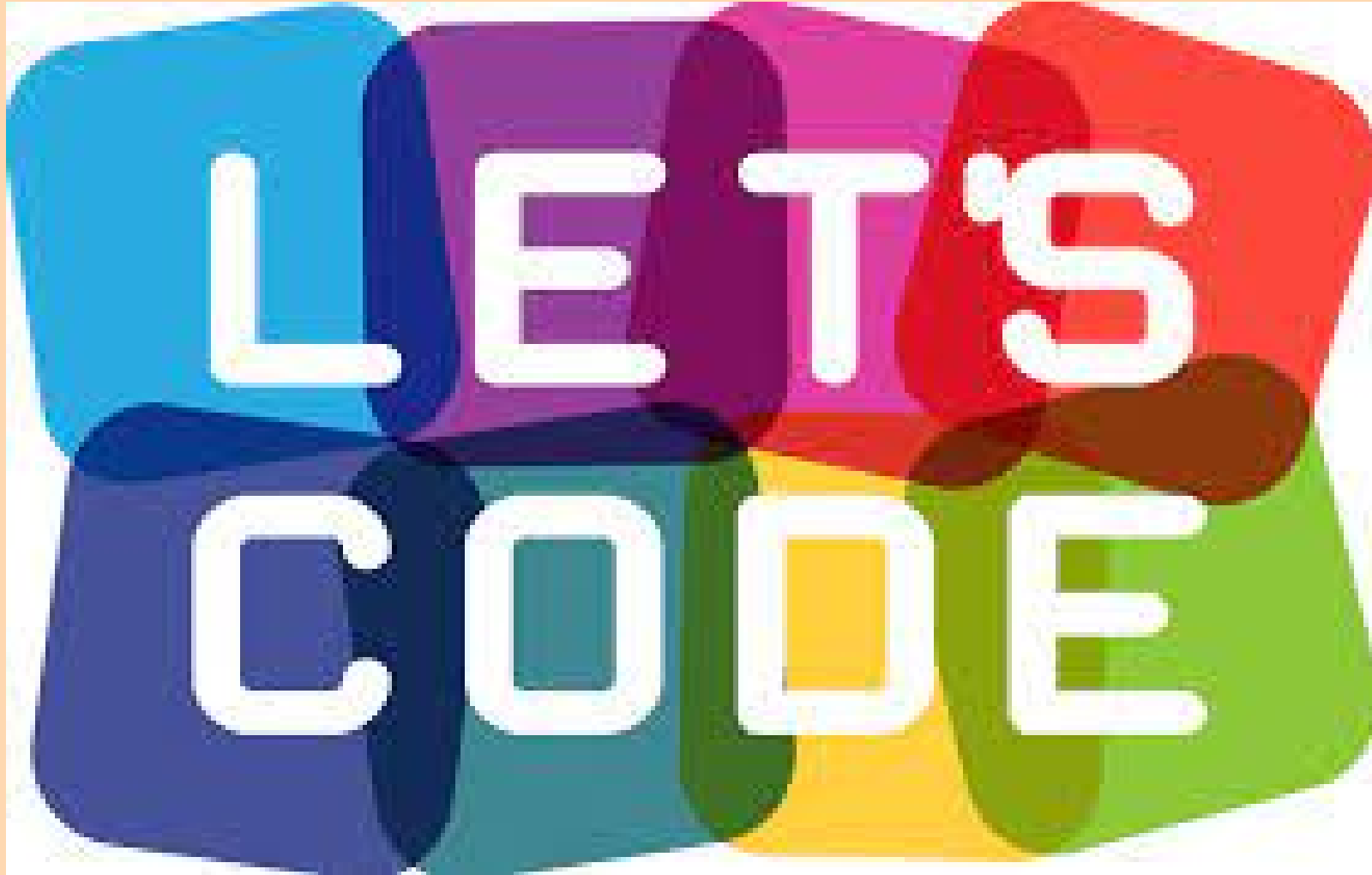


How does **the Mirror** work? (**stairs.c**)

# What are the smallest sub-problems?



## Solve them without recursion!



[min-re.c](#)   [gcd-re.c](#)   [bsearch-re.c](#)   [mergesort.c](#)



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

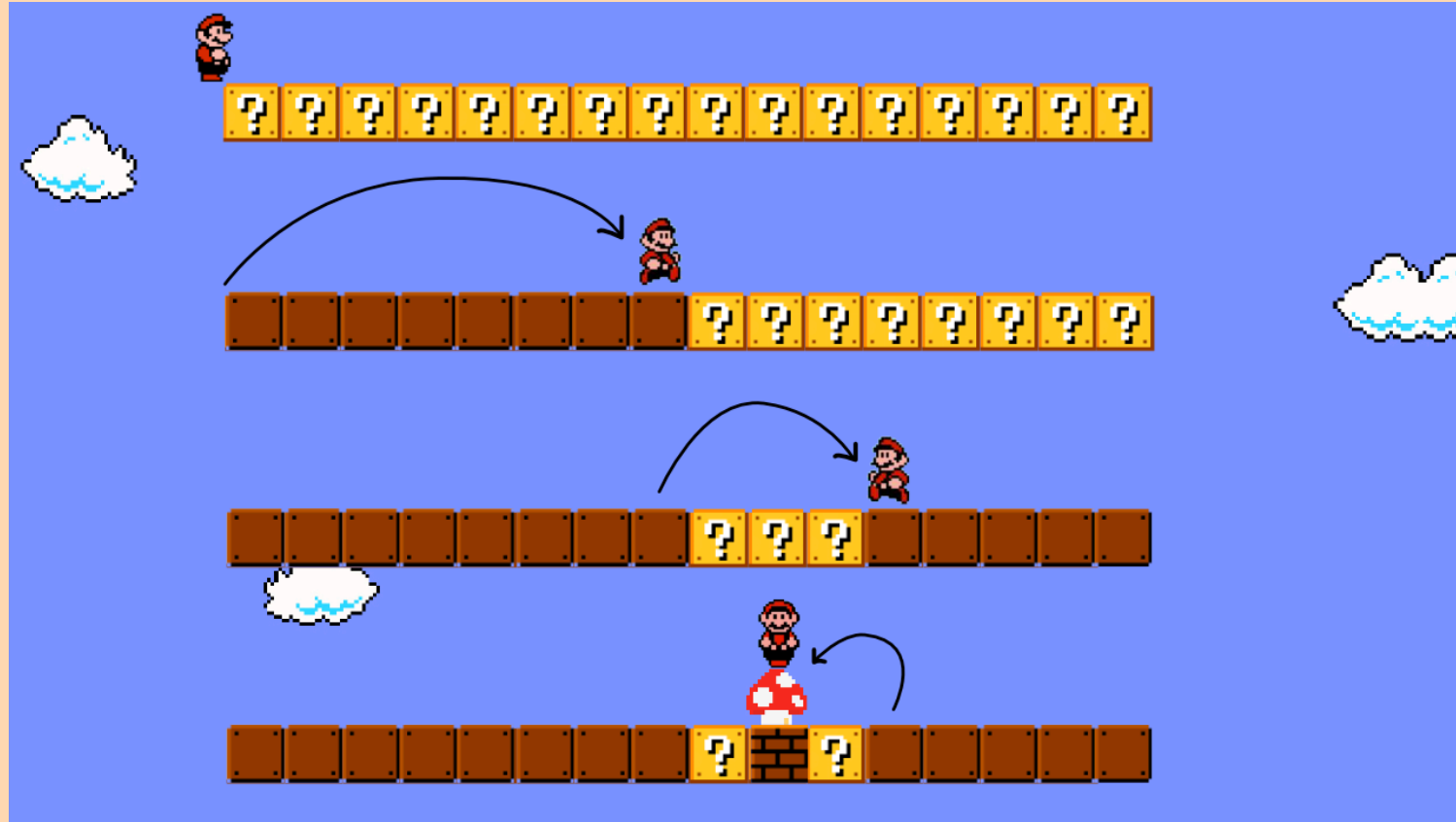
$$\begin{aligned}\text{Min}(3, 5, 2, 7) &= \min(7, \text{Min}(3, 5, 2)) \\ &= \min(7, \min(2, \text{Min}(3, 5))) \\ &= \min(7, \min(2, \min(5, \text{Min}(3)))) \\ &= \min(7, \min(2, \min(5, 3))) \\ &= \min(7, \min(2, 3)) \\ &= \min(7, 2) \\ &= 2\end{aligned}$$

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

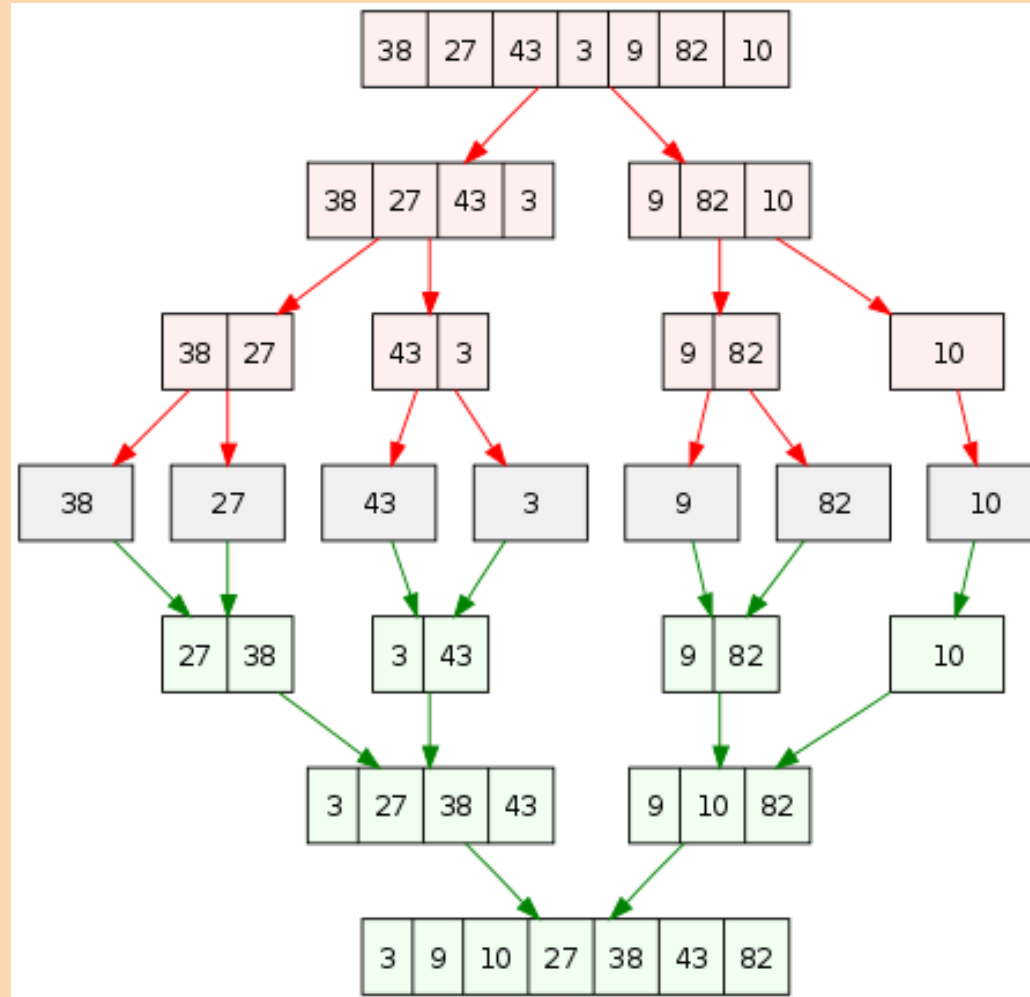


$$\text{gcd}(a, b) = \text{gcd}(b, a \% b)$$


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



# MergeSort (**mergesort.c**)







Happy End  
of Semester



