Algorithm Design

IGCSE Computer Science ▪ Topic 7

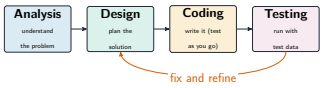
IGCSE Computer Science

What we will cover

1. The development life cycle
2. Design tools
3. Algorithms & IPO
4. Validation
5. Trace tables
6. Search & sort

1 Designing

The development life cycle



- **abstraction** 抽象 – keep only the important details and ignore the rest;
- **decomposition** 分解 – break a big problem into smaller, easier parts.

Three design tools

structure diagram	**flowchart** 流程图	**pseudocode** 伪代码
the parts of a system and how they fit	boxes and arrows for the steps in order	steps in simple, code-like English

> None is a real language – they plan the solution before any code is written.

Algorithms: input, process, output

An **algorithm** 算法 is an ordered set of steps that solves a problem. Every one splits into:

input → process → output

> "Average of three marks": input the marks, process add and divide by 3, output the average. Name all three in an exam answer.

2 Checking

Validation checks

range 范围 – between a low and high value

length 长度 – allowed number of characters

type 类型 – the right kind (number, not letters)

presence 存在性 – something was entered

format 格式 – the right pattern (dd/mm/yyyy)

> Validation checks data is *sensible*; verification (visual or double-entry) checks it was *copied right*.

Worked example – a trace table

Trace with input $n = 5$. What does it output?

```
total <- 0
FOR i <- 1 TO n
total <- total + i
NEXT i
OUTPUT total
```

i	total	OUT
1	1	
2	3	
3	6	
4	10	
5	15	15

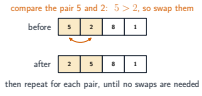
> It sums 1 to n: output 15. Fill *one row per pass* – never run the loop in your head.

3 Search and sort

Linear search and bubble sort

search for 5: check each item in turn (left to right)

linear search – check each in turn



- **linear search** 线性查找: check every item until found
- **bubble sort** 冒泡排序: compare neighbours, swap, repeat until no swaps

> Linear search works on any list; bubble sort finishes when a whole pass makes no swaps.

4 Recap

Topic 7 – key takeaways

1. **Design**
   abstraction + decomposition; 3 design tools
2. **IPO**
   every algorithm: input, process, output
3. **Check**
   validation = sensible; verification = copied right
4. **Trace**
   one row per pass to see what code does

Check yourself

1. What are the three parts of every algorithm?
2. Range check or type check for "age 0–120"?
3. Validation or verification: double entry?
4. When does a bubble sort stop?

Answers **1.** input, process, output **2.** range **3.** verification **4.** when a pass makes no swaps