

Thuật toán

Tin học · Lớp 10

Giáo viên: Phúc Hảo

- ▶ Hiểu được khái niệm **thuật toán** và vai trò của nó trong giải quyết vấn đề.
- ▶ Nắm được các **tính chất** cơ bản mà một thuật toán cần có.
- ▶ Phân biệt và mô tả được **ba cấu trúc điều khiển**: tuần tự, rẽ nhánh, lặp.
- ▶ Biết dùng **lưu đồ** để biểu diễn thuật toán đơn giản.

Ý chính: Mọi việc làm theo trình tự đều là một "thuật toán" đời thường!

Buổi sáng em pha một cốc mì gói: *đun nước sôi* → *cho mì và gói gia vị* → *chờ 3 phút* → *thưởng thức*. Nếu đảo lộn thứ tự, kết quả sẽ khác hẳn!

Câu hỏi gợi mở

Làm thế nào để **ra lệnh cho máy tính** thực hiện một công việc, khi máy tính không "hiểu" như con người? Ta cần mô tả công việc thành các **bước rõ ràng, chính xác** — đó chính là **thuật toán**.

1

Khái niệm thuật toán

Thuật toán là gì?

Định nghĩa

Thuật toán là một dãy hữu hạn các **thao tác được sắp xếp theo trình tự xác định**, sao cho khi thực hiện dãy thao tác đó với **đầu vào** (Input) sẽ thu được **kết quả** (Output) mong muốn.

- ▶ **Input:** thông tin đưa vào (ví dụ: số cần kiểm tra).
- ▶ **Output:** kết quả nhận được (ví dụ: "số nguyên tố" hoặc "không phải").

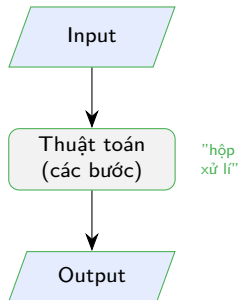
Liên hệ thực tế

Công thức nấu ăn là một thuật toán: **nguyên liệu** (Input) → các bước chế biến → **món ăn** (Output).

Vì sao cần thuật toán?

- ▶ Máy tính chỉ làm được việc khi được **chỉ dẫn từng bước một cách rõ ràng**.
- ▶ Thuật toán giúp ta **tách vấn đề lớn** thành các thao tác nhỏ, dễ kiểm soát.
- ▶ Một thuật toán tốt cho **cùng một kết quả** dù ai thực hiện.

Ý chính: Viết chương trình = mô tả thuật toán bằng ngôn ngữ máy hiểu.



Ghi nhớ — 5 tính chất

- ▶ **Tính dừng:** luôn kết thúc sau hữu hạn bước.
- ▶ **Tính xác định:** mỗi bước rõ ràng, không mập mờ.
- ▶ **Tính đúng đắn:** cho ra kết quả đúng với mọi Input hợp lệ.
- ▶ **Tính đầu vào — đầu ra:** có Input và Output rõ ràng.
- ▶ **Tính khả thi:** thực hiện được, không đòi hỏi vô hạn tài nguyên.

Lỗi thường gặp

Một dãy bước "lặp mãi không dừng" **không phải thuật toán** vì vi phạm **tính dừng**.

Có **ba cách** phổ biến để mô tả một thuật toán:

- ▶ **Liệt kê các bước** bằng ngôn ngữ tự nhiên.
- ▶ **Lưu đồ (sơ đồ khối)**: dùng hình khối trực quan.
- ▶ **Mã giả**: gần với ngôn ngữ lập trình.

Ý chính: Lưu đồ giúp "nhìn thấy" luồng xử lý một cách trực quan nhất.

term: Bắt đầu/Kết thúc

io: Nhập/Xuất

proc: Xử lý

dec: Rẽ nhánh?

2

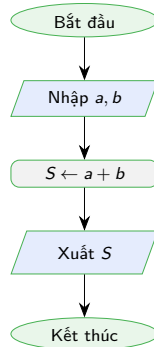
Ba cấu trúc điều khiển

Tuần tự

Các thao tác được thực hiện **lần lượt**, theo đúng **thứ tự** từ trên xuống, mỗi bước một lần.

Liên hệ thực tế

Mở cửa → bật đèn → ngồi vào bàn → mở sách. Không thể mở sách trước khi ngồi vào bàn!



Rẽ nhánh

Dựa vào một **điều kiện**, thuật toán chọn thực hiện nhánh **Đúng** hoặc nhánh **Sai**. Có dạng **thiếu** (chỉ làm khi Đúng) và dạng **đủ** (Đúng làm A, Sai làm B).

Liên hệ thực tế

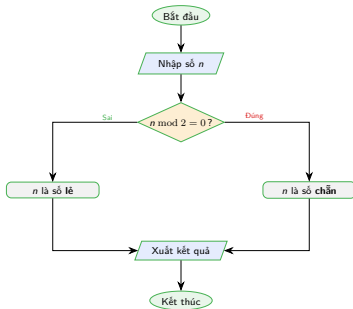
"*Nếu trời mưa thì mang ô, ngược lại đội mũ.*" — điều kiện quyết định hành động.

Ghi nhớ

Trong lưu đồ, khối rẽ nhánh vẽ hình **thoi** (dec), có hai nhánh gắn nhãn **Đúng** / **Sai**.

Lưu đồ minh họa rẽ nhánh: số chẵn hay lẻ

Lưu đồ: kiểm tra số chẵn hay lẻ



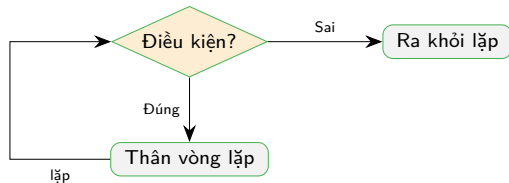
Lặp

Một nhóm thao tác được **thực hiện lặp đi lặp lại** khi điều kiện còn thoả mãn.

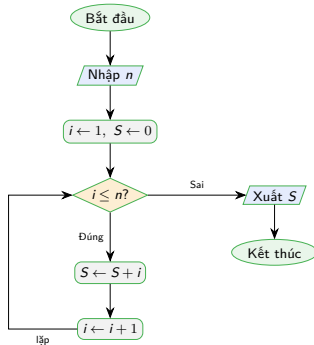
- ▶ **Lặp với số lần biết trước** (for).
- ▶ **Lặp với điều kiện** (while) — chưa biết trước số lần.

Liên hệ thực tế

"Rót nước vào bình *cho đến khi đầy*"
— lặp cho tới khi điều kiện dừng.



Lưu đồ vòng lặp: tính tổng $S = 1 + 2 + \dots + n$



Cấu trúc	Đặc điểm	Ví dụ đời thường
Tuần tự	Làm lần lượt từng bước	Đánh răng → rửa mặt
Rẽ nhánh	Chọn nhánh theo điều kiện	Mưa thì mang ô
Lặp	Làm lại nhiều lần	Chép phạt 10 lần

Ý chính: Mọi thuật toán, dù phức tạp, đều được ghép từ **ba cấu trúc** này.

3

Ví dụ vận dụng

Ví dụ 1: Tìm số lớn nhất của hai số a, b

Đề bài

Cho hai số a và b . Hãy mô tả thuật toán tìm số lớn nhất, gọi là Max .

Bước 1. *Xác định Input/Output.* Input: hai số a, b ; Output: Max .

Đây là bài toán so sánh $\Rightarrow$ cần cấu trúc rẽ nhánh.

Bước 2. *Đặt điều kiện so sánh.* Kiểm tra $a > b$?

Nếu $a > b$ (Đúng) $\Rightarrow Max \leftarrow a$.

Bước 3. *Xử lý nhánh Sai.* Ngược lại thì b lớn hơn hoặc bằng a :

Nếu $a > b$ Sai $\Rightarrow Max \leftarrow b$.

Bước 4. *Xuất kết quả.* In ra giá trị Max rồi kết thúc.

Kết quả

Thuật toán luôn dừng sau hữu hạn bước và cho ra $Max = \max(a, b)$.

Ví dụ 2: Đếm số lần lặp để chia hết cho 2

Đề bài

Cho số nguyên dương n . Đếm xem phải chia n cho 2 (lấy phần nguyên) **bao nhiêu lần** thì n về 1.

Bước 1. *Khởi tạo.* Đặt biến đếm $d \leftarrow 0$. Vì chưa chia lần nào.

$$d \leftarrow 0.$$

Bước 2. *Điều kiện lặp.* Kiểm tra $n > 1$? Đây là lặp `while` vì chưa biết trước số lần.

Khi $n > 1$ (Đúng): thực hiện thân vòng lặp.

Bước 3. *Thân vòng lặp.* Chia đôi n và tăng biến đếm:

$$n \leftarrow \lfloor n/2 \rfloor, \quad d \leftarrow d + 1.$$

Bước 4. *Kết thúc lặp & xuất.* Khi $n \leq 1$ (Sai) thì thoát và xuất d .

Kết quả

Với $n = 8$: $8 \rightarrow 4 \rightarrow 2 \rightarrow 1$, cần $d = 3$ lần chia.

Luyện tập nhanh · Nhận biết

Thuật toán là gì?

- A. Một ngôn ngữ lập trình dùng để viết chương trình
- B. Một dãy hữu hạn các bước thao tác để giải quyết một bài toán
- C. Một phần mềm chạy trên máy tính
- D. Một thiết bị phần cứng xử lý dữ liệu

Đáp án đúng: **B. Một dãy hữu hạn các bước thao tác để giải quyết một bài toán.**

Giải thích: Thuật toán là các *bước*, không phải ngôn ngữ (A – nhầm với công cụ diễn đạt), không phải phần mềm (C) hay phần cứng (D – nhầm giữa cách giải và phương tiện thực hiện).

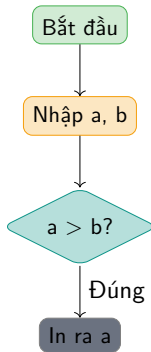
Luyện tập nhanh · Nhận biết

Trong thuật toán, cấu trúc nào cho phép *lặp lại* một nhóm thao tác nhiều lần?

- A. Cấu trúc tuần tự
- B. Cấu trúc rẽ nhánh
- C. Cấu trúc lặp
- D. Cấu trúc nhập/xuất

Đáp án đúng: **C. Cấu trúc lặp.**

Giải thích: Lặp = thực hiện lại. Tuần tự (A) chỉ chạy thẳng một lần; rẽ nhánh (B) là chọn theo điều kiện – dễ nhầm vì cả hai đều có "điều kiện"; D không phải cấu trúc điều khiển.



Luyện tập nhanh · Nhận biết

Hình bên là sơ đồ khối. Khối hình thoi biểu diễn thao tác nào?

- A. Bắt đầu / kết thúc
- B. Nhập / xuất dữ liệu
- C. Kiểm tra điều kiện (rẽ nhánh)
- D. Thao tác tính toán

Đáp án đúng: **C. Kiểm tra điều kiện (rẽ nhánh).**

Giải thích: Hình thoi luôn là điều kiện. Hình bo tròn là bắt đầu/kết thúc (A), hình bình hành là nhập/xuất (B), hình chữ nhật là tính toán (D) – học sinh hay nhầm hình thoi với hình chữ nhật.

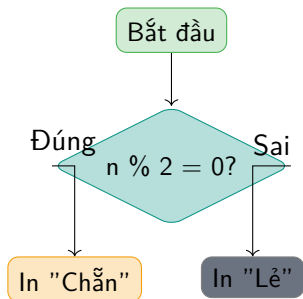
Luyện tập nhanh · Thông hiểu

Cho đoạn thuật toán: $S = 0$; $i = 1$; lặp *khi* $i \leq 3$: $S = S + i$; $i = i + 1$. Sau khi kết thúc, giá trị của S là bao nhiêu?

- A. 3
- B. 6
- C. 10
- D. 4

Đáp án đúng: **B. 6.**

Giải thích: $S = 1 + 2 + 3 = 6$. Đáp án A (=3) quên cộng dồn chỉ lấy i cuối; C (=10) lặp thêm cả $i = 4$ vì đọc điều kiện là $i < 4$ thành $i \leq 4$; D (=4) tính nhầm $1 + 3$.



Luyện tập nhanh · Thông hiểu

Với sơ đồ bên, nếu nhập $n = 7$ thì kết quả in ra là gì?

- A. "Chẵn"
- B. "Lẻ"
- C. Cả "Chẵn" và "Lẻ"
- D. Không in gì cả

Đáp án đúng: **B. "Lẻ"**.

Giải thích: $7\%2 = 1 \neq 0$ nên điều kiện Sai $\rightarrow$ nhánh phải. A sai vì đọc nhầm điều kiện; C sai vì rẽ nhánh chỉ đi *một* đường; D sai vì mỗi nhánh đều có lệnh in.

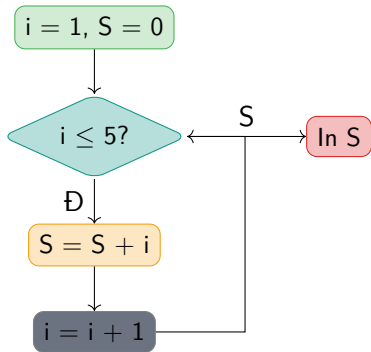
Luyện tập nhanh · Thông hiểu

Tính chất "*tính dừng*" của thuật toán có nghĩa là gì?

- A. Thuật toán phải kết thúc sau hữu hạn bước
- B. Thuật toán luôn cho cùng một kết quả
- C. Thuật toán được viết ngắn gọn nhất có thể
- D. Thuật toán chạy càng nhanh càng tốt

Đáp án đúng: **A. Thuật toán phải kết thúc sau hữu hạn bước.**

Giải thích: Tính dừng = chắc chắn kết thúc. B là tính *xác định*; D là tính *hiệu quả* – học sinh hay lẫn các tính chất với nhau; C không phải yêu cầu bắt buộc.



Luyện tập nhanh · Vận dụng

Đọc sơ đồ vòng lặp bên. Giá trị S được in ra là bao nhiêu?

- A. 10
- B. 15
- C. 21
- D. 25

Đáp án đúng: **B. 15.**

Giải thích: $S = 1 + 2 + 3 + 4 + 5 = 15$. A (=10) dừng sớm ở $i = 4$; C (=21) lặp thừa tới $i = 6$ do đọc $i \leq 5$ thành $i \leq 6$; D (=25) nhầm với 5×5 .

Luyện tập nhanh · Vận dụng

Cần mô tả thuật toán: "Nếu điểm ≥ 5 thì đạt, ngược lại thì học lại". Kết hợp cấu trúc điều khiển nào là phù hợp nhất?

- A. Chỉ dùng cấu trúc tuần tự
- B. Chỉ dùng cấu trúc lặp
- C. Cấu trúc rẽ nhánh dạng đủ (nếu... thì... ngược lại...)
- D. Cấu trúc rẽ nhánh dạng thiếu (chỉ có nếu... thì...)

Đáp án đúng: **C. Rẽ nhánh dạng đủ.**

Giải thích: Có *hai* khả năng (đạt / học lại) nên cần dạng đủ. D dạng thiếu sẽ bỏ sót trường hợp "ngược lại" – lỗi hay gặp; A, B không xử lí được việc chọn theo điều kiện.

1. Viết thuật toán (liệt kê bước) tìm số lớn nhất trong ba số a, b, c .
2. Vẽ sơ đồ khối kiểm tra một số n nhập vào là số dương, số âm hay bằng 0.
3. Mô tả thuật toán tính tổng $S = 1 + 2 + \dots + n$ bằng cấu trúc lặp.

Gợi ý

Bài 1: dùng rẽ nhánh, so sánh từng cặp và giữ lại giá trị lớn hơn (biến `max`). Bài 2: cần rẽ nhánh *nhiều nhánh*. Bài 3: khởi tạo $S = 0$, dùng biến đếm i chạy từ 1 đến n và cộng dồn.

1. Viết thuật toán giải phương trình bậc nhất $ax + b = 0$ (xét mọi trường hợp của a, b).
2. Vẽ sơ đồ khối tính giai thừa $n! = 1 \cdot 2 \cdot \dots \cdot n$.
3. Mô tả thuật toán đếm số chẵn trong dãy n số nhập vào.
4. Viết thuật toán kiểm tra một số n có phải số nguyên tố hay không.
5. Cho $S = 2 + 4 + 6 + \dots + 100$: mô tả bằng cấu trúc lặp và tính kết quả.

Đọc trước bài sau

Đọc trước bài "**Ngôn ngữ lập trình bậc cao và ngôn ngữ Python**" – tìm hiểu cách biến thuật toán thành chương trình chạy được.

Tóm tắt: Ba cấu trúc điều khiển

Cấu trúc	Ý nghĩa	Ký hiệu sơ đồ
Tuần tự	Thực hiện lần lượt từng bước	Chuỗi hình chữ nhật
Rẽ nhánh	Chọn hướng theo điều kiện	Hình thoi (điều kiện)
Lặp	Lặp lại nhóm thao tác	Vòng lặp về điều kiện

Hình khối	Chức năng
Bo tròn	Bắt đầu / Kết thúc
Hình bình hành	Nhập / Xuất dữ liệu
Hình chữ nhật	Xử lý, tính toán
Hình thoi	Kiểm tra điều kiện

- ▶ **Thuật toán** là dãy hữu hạn bước để giải một bài toán, có tính dừng, xác định và đúng đắn.
- ▶ Có **ba cấu trúc điều khiển** cơ bản: tuần tự, rẽ nhánh, lặp – kết hợp linh hoạt để mô tả mọi thuật toán.
- ▶ **Sơ đồ khối** giúp trình bày thuật toán trực quan; cần nhớ đúng ý nghĩa từng hình khối.
- ▶ Khi lặp, luôn chú ý **điều kiện dừng** để tránh lặp vô hạn hoặc lặp thiếu/thừa.

Liên hệ

Mọi ứng dụng em dùng hằng ngày – từ tìm đường trên bản đồ đến gợi ý video – đều bắt đầu từ một thuật toán được diễn đạt bằng chính ba cấu trúc điều khiển này.

Cảm ơn các em đã lắng nghe!

Bài giảng biên soạn với EduFlow · agentra.io.vn
