

## §6. GIẢI BÀI TOÁN TRÊN MÁY TÍNH

---

Học sử dụng máy tính thực chất là học cách giao cho máy tính việc mà ta muốn nó làm. Khả năng khai thác máy tính phụ thuộc rất nhiều vào sự hiểu biết của người dùng. Việc giải bài toán trên máy tính thường được tiến hành qua các bước sau:

*Bước 1.* Xác định bài toán;

*Bước 2.* Lựa chọn hoặc thiết kế thuật toán;

*Bước 3.* Viết chương trình;

*Bước 4.* Hiệu chỉnh;

*Bước 5.* Viết tài liệu.

### 1. Xác định bài toán

Như đã trình bày, mỗi bài toán được đặc tả bởi hai thành phần: Input và Output. Việc xác định bài toán chính là xác định rõ hai thành phần này và mối quan hệ giữa chúng. Các thông tin đó cần được nghiên cứu cẩn thận để có thể lựa chọn thuật toán, cách thể hiện các đại lượng đã cho, các đại lượng phát sinh trong quá trình giải bài toán và ngôn ngữ lập trình thích hợp.

Ví dụ, trong một bài toán tin học khi đề cập đến một số nguyên dương  $N$ , là tuổi của một người, có thể chỉ rõ phạm vi giá trị của  $N$  từ 1 đến 150, để lựa chọn cách thể hiện  $N$  bằng kiểu dữ liệu thích hợp.

### 2. Lựa chọn hoặc thiết kế thuật toán

#### a) *Lựa chọn thuật toán*

Bước lựa chọn hoặc thiết kế thuật toán là bước quan trọng nhất để giải một bài toán.

Mỗi thuật toán chỉ giải một bài toán nào đó, nhưng có thể có nhiều thuật toán khác nhau cùng giải một bài toán. Cần thiết kế hoặc chọn một thuật toán phù hợp đã có để giải bài toán cho trước.

Khi thiết kế hoặc lựa chọn thuật toán người ta thường quan tâm đến các tài nguyên như thời gian thực hiện, số lượng ô nhớ,... Trong các loại tài nguyên, người ta quan tâm nhiều nhất đến thời gian vì đó là dạng tài nguyên không tái tạo được.

Ví dụ, với bài toán tìm kiếm, nếu dãy đã cho là dãy đã sắp xếp thì dễ thấy thuật toán tìm kiếm nhị phân cần ít thao tác hơn nhiều so với thuật toán tìm kiếm tuần tự. Vì thế nó cần ít thời gian thực hiện hơn.

Một tiêu chí khác được rất nhiều người quan tâm là cần thiết kế hoặc lựa chọn thuật toán sao cho việc viết chương trình cho thuật toán đó ít phức tạp.

Khi thiết kế hoặc lựa chọn thuật toán để giải một bài toán cụ thể cần căn cứ vào lượng tài nguyên mà thuật toán đòi hỏi và lượng tài nguyên thực tế cho phép.

#### ***b) Diễn tả thuật toán***

Việc diễn tả một thuật toán đã được trình bày ở §4. Dưới đây ta xét thêm một ví dụ khác.

**Ví dụ. Tìm ước chung lớn nhất (UCLN) của hai số nguyên dương  $M$  và  $N$ .**

##### **• Xác định bài toán**

– *Input*: Cho  $M, N$ ;

– *Output*:  $UCLN(M, N)$ .

• **Ý tưởng**: Sử dụng những điều đã biết sau:

– Nếu  $M = N$  thì giá trị chung đó là UCLN của  $M$  và  $N$ ;

– Nếu  $M < N$  thì  $UCLN(M, N) = UCLN(M, N - M)$ ;

– Nếu  $M > N$  thì  $UCLN(M, N) = UCLN(M - N, N)$ .

##### **• Thuật toán**

*Cách liệt kê*

*Bước 1.* Nhập  $M, N$ ;

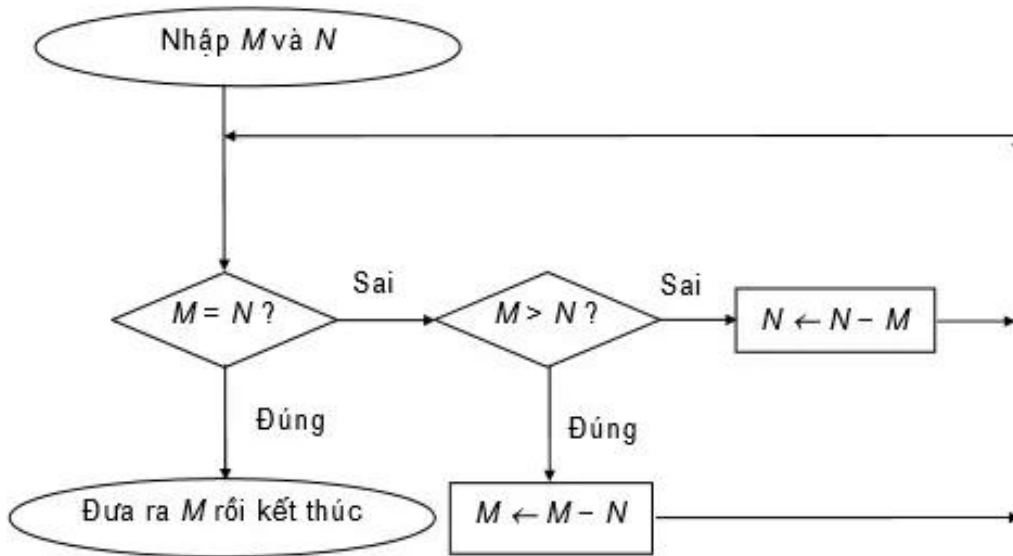
*Bước 2.* Nếu  $M = N$  thì lấy giá trị chung này làm UCLN rồi chuyển đến bước 5;

*Bước 3.* Nếu  $M > N$  thì  $M \leftarrow M - N$  rồi quay lại bước 2;

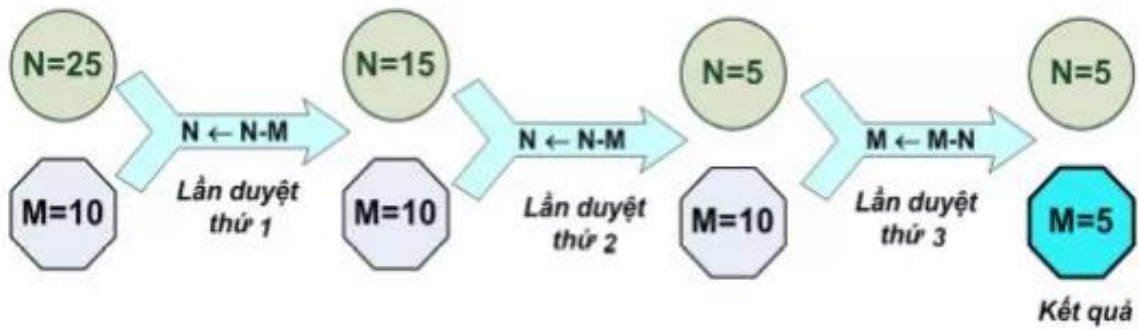
*Bước 4.*  $N \leftarrow N - M$  rồi quay lại bước 2;

*Bước 5.* Đưa ra kết quả UCLN rồi kết thúc.

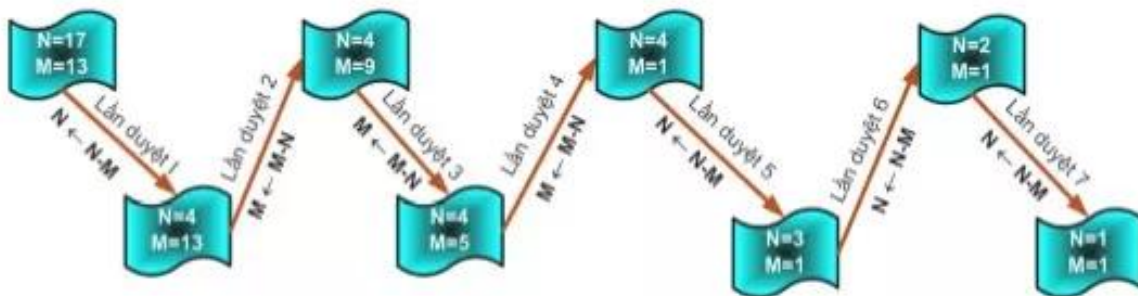
Sơ đồ khối



Sau đây là hai ví dụ mô phỏng việc thực hiện thuật toán trên.



a)  $UCLN(25, 10) = 5$



b)  $UCLN(17, 13) = 1$

### 3. Viết chương trình

Việc viết chương trình là tổng hợp giữa việc lựa chọn cách tổ chức dữ liệu và sử dụng ngôn ngữ lập trình để diễn đạt đúng thuật toán.

Khi viết chương trình ta nên chọn một ngôn ngữ lập trình hoặc một phần mềm chuyên dụng thích hợp với thuật toán. Viết chương trình trong ngôn ngữ nào thì cần phải tuân theo đúng quy định ngữ pháp của ngôn ngữ đó.

Chương trình dịch chỉ có thể phát hiện và thông báo các lỗi về mặt ngữ pháp.

### 4. Hiệu chỉnh

Sau khi được viết xong, chương trình vẫn còn có thể có nhiều lỗi khác chưa phát hiện được nên có thể không cho kết quả đúng. Vì vậy, cần phải thử chương trình bằng cách thực hiện nó với một số bộ Input tiêu biểu phụ thuộc vào đặc thù của bài toán và bằng cách nào đó ta đã biết trước Output. Các bộ Input và Output tương ứng này được gọi là các Test. Nếu có sai sót, ta phải sửa chương trình rồi thử lại. Quá trình này được gọi là *hiệu chỉnh*. Nếu kết quả hiệu chỉnh cho thấy ngôn ngữ lập trình hoặc thuật toán không phù hợp, thậm chí ta phải quay lại lựa chọn hay thiết kế thuật toán.

Ví dụ, để kiểm chứng tính đúng đắn của chương trình giải phương trình bậc hai  $ax^2 + bx + c = 0$ , ta có thể sử dụng ba bộ Input như sau:

- Biệt số  $\Delta > 0$ :  $a = 1, b = -5, c = 6$  (chương trình đưa ra hai nghiệm);
- Biệt số  $\Delta = 0$ :  $a = 1, b = -4, c = 4$  (chương trình đưa ra một nghiệm);
- Biệt số  $\Delta < 0$ :  $a = 1, b = 4, c = 8$  (chương trình thông báo phương trình vô nghiệm).

### 5. Viết tài liệu

Tài liệu phải mô tả bài toán, thuật toán, thiết kế chương trình, kết quả thử nghiệm và hướng dẫn sử dụng. Tài liệu này rất có ích cho người sử dụng chương trình và cho việc đề xuất những khả năng hoàn thiện thêm.

Các bước trên có thể lặp đi lặp lại nhiều lần cho đến khi ta cho rằng chương trình đã làm việc đúng đắn và hiệu quả.

## **Các thuật ngữ chính**

---

Tài nguyên; Viết chương trình; Hiệu chỉnh; Viết tài liệu.

---

## **CÂU HỎI VÀ BÀI TẬP**

1. Hãy nêu tiêu chuẩn lựa chọn thuật toán.
2. Hãy nêu nội dung và mục đích của bước hiệu chỉnh khi giải bài toán trên máy tính.
3. Hãy viết thuật toán giải phương trình  $ax + b = 0$  và đề xuất các Test tiêu biểu.