

Bài 1: THUẬT TOÁN VÀ CÁCH BIỂU DIỄN

THUẬT TOÁN

I. THUẬT TOÁN:

Khái niệm: Trong tin học, thuật toán là sự mô tả phương pháp giải bài toán mà thích hợp cho việc cài đặt thành chương trình nhờ các ngôn ngữ lập trình. Một thuật toán thường được thể hiện bởi một thủ tục gồm một dãy hữu hạn bước mà theo đó sẽ đạt đến lời giải bài toán.

Ví dụ 1: Trình bày thuật toán giải phương trình $ax+b=0$

II. BIỂU DIỄN THUẬT TOÁN

Để trình bày một thuật toán hay biểu diễn một thuật toán, ta có thể dùng ngôn ngữ tự nhiên, dùng lưu đồ hay dùng mã giả.

1. Dùng ngôn ngữ tự nhiên (*natural language*)

Trong cách sử dụng ngôn ngữ tự nhiên ta sẽ liệt kê các bước thực hiện các thao tác hay công việc nào đó của thuật toán bằng ngôn ngữ mà con người sử dụng một cách phổ thông hàng ngày.

Ví dụ 2: Biểu diễn thuật toán giải phương trình $ax+b=0$ bằng ngôn ngữ tự nhiên

Bước 1: Nhập a, b

Bước 2: Nếu $a=0$

Nếu $b=0$ xuất “Phương trình vô số nghiệm”

Nếu $b \neq 0$ xuất “Phương trình vô nghiệm”

Nếu $a \neq 0$ xuất “Phương trình có nghiệm $x = \frac{-b}{a}$ ”

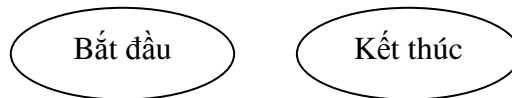
Nhận xét : Mặc dù cách biểu diễn này khá tự nhiên và không đòi hỏi người viết thuật toán phải biết nhiều quy ước khác, nhưng nó không thể hiện tính cấu trúc của thuật toán nên không thuận lợi cho việc thiết kế và cài đặt những thuật toán phức tạp. Hơn nữa trong nhiều trường hợp việc biểu diễn thuật toán bằng ngôn ngữ tự nhiên tỏ ra dài dòng và sẽ gây ra sự nhầm lẫn đối với người đọc.

2. Dùng lưu đồ (*flow chart*):

Ngôn ngữ lưu đồ hay sơ đồ khối là một công cụ rất trực quan để diễn đạt các thuật toán. Biểu diễn bằng lưu đồ sẽ giúp ta có được một cái nhìn tổng quan về toàn cảnh của quá trình xử lý theo thuật toán.

Lưu đồ là một hệ thống các nút có hình dạng khác nhau, thể hiện các chức năng khác nhau và được nối với nhau bởi các cung. Lưu đồ được tạo thành bởi 4 thành phần chủ yếu sau đây:

Nút giới hạn: được biểu diễn bởi hình ôvan có ghi chữ bên trong như :

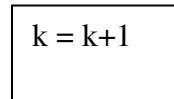


Các nút trên còn được gọi là nút đầu và nút cuối của lưu đồ.

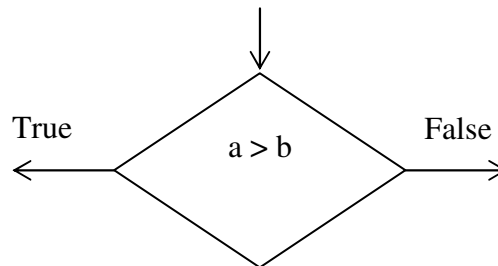
Nút nhập/xuất: một hình bình hành có ghi các giá trị cần nhập.



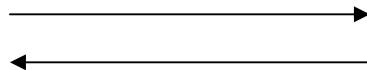
Nút thao tác: một hình chữ nhật có ghi các lệnh cần thực hiện.



Nút điều kiện: thường là một hình thoi có ghi điều kiện cần kiểm tra. Trong các cung nối với nút này có 2 cung ra chỉ hướng đi theo 2 trường hợp: điều kiện đúng và điều kiện sai.

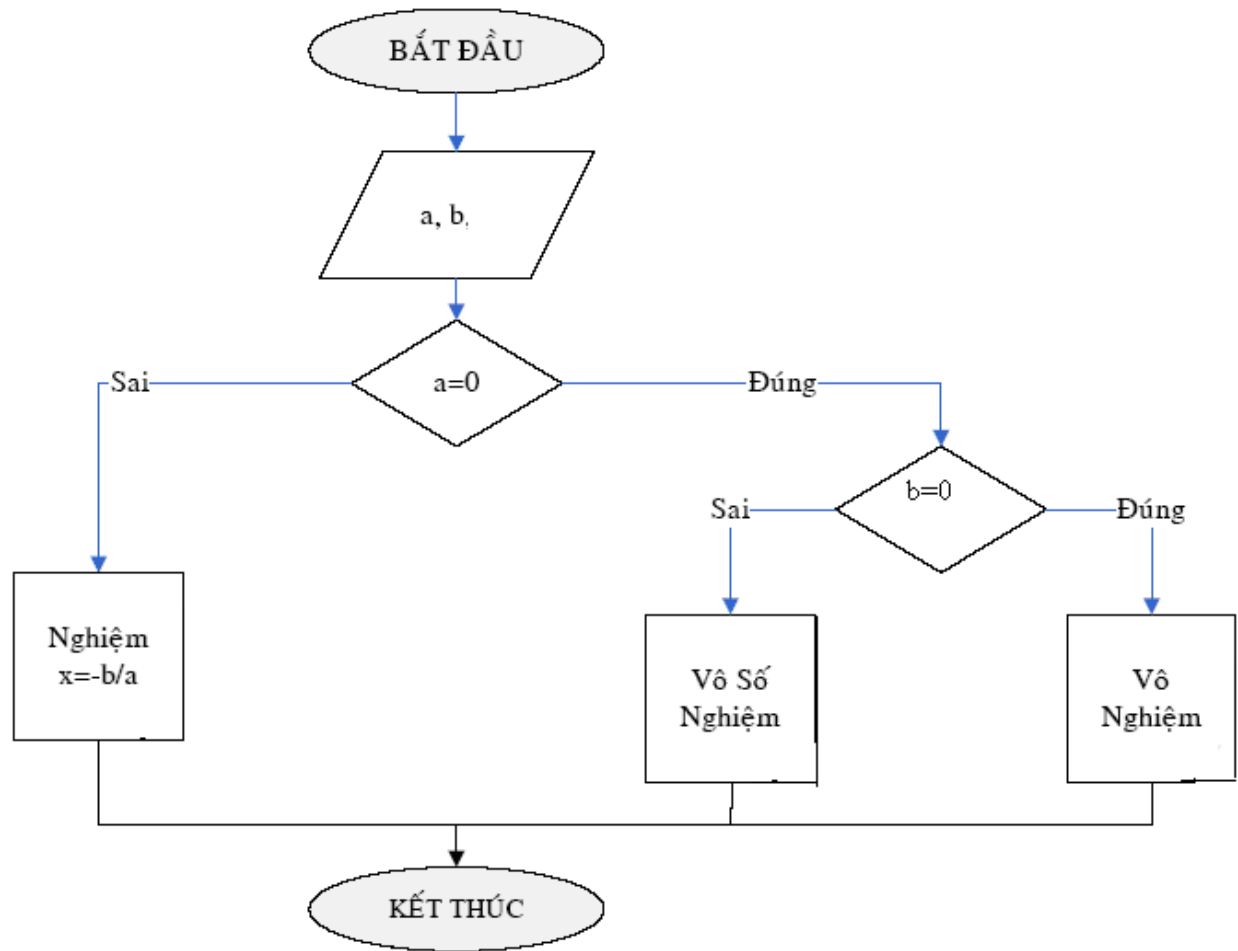


Cung: là các đường nối từ nút này đến nút khác của lưu đồ.



Hoạt động của thuật toán theo lưu đồ được bắt đầu từ nút đầu tiên. Sau khi thực hiện các thao tác hoặc kiểm tra điều kiện ở mỗi nút thì bộ xử lý sẽ theo một cung để đến nút khác. Quá trình thực hiện thuật toán dừng khi gặp nút kết thúc hay nút cuối.

Ví dụ 3: Biểu diễn thuật toán giải phương trình $ax+b=0$ bằng lưu đồ



Nhận xét : Việc sử dụng lưu đồ sẽ rất cồng kềnh đối với các thuật toán phức tạp.

3. Dùng mã giả (*pseudocode*)

Để biểu diễn thuật toán một cách hiệu quả, người ta thường dùng mã giả. Theo cách này, ta sẽ sử dụng một số qui ước của một ngôn ngữ lập trình, chẳng hạn là ngôn ngữ lập trình PASCAL hay C, nhất là các cấu trúc điều khiển của ngôn ngữ lập trình như các cấu trúc chọn, các cấu trúc lặp.

Trong mã giả ta còn sử dụng cả các ký hiệu toán học, các biến, và đôi khi cả cấu trúc kiểu thủ tục. Cấu trúc thuật toán kiểu thủ tục thường được sử dụng để trình bày các thuật toán đệ qui hay các thuật toán quá phức tạp cần phải được trình bày thành nhiều cấp độ.

Cùng với việc sử dụng các biến, trong thuật toán rất thường gặp một phát biểu hành động đặt (hay gán) một giá trị cho một biến. Ví dụ hành động tăng biến i lên 1 có thể được viết như sau:

$$i = i + 1$$

Các cấu trúc thường được sử dụng trong mã giả dựa theo ngôn ngữ lập trình C gồm:

1/ Cấu trúc chọn:

*** Cấu trúc if**

if (điều kiện)

(hành động)

*** Cấu trúc if.. else..**

if (điều kiện)

(hành động)

else (hành động)

2/ Cấu trúc lặp:

*** Cấu trúc while**

while (điều kiện)

{

(hành động)

}

*** Cấu trúc for**

for (i = giá trị đầu; điều kiện lặp; i+=giá trị bước nhảy)

{

(hành động)

}

Ví dụ 4: Biểu diễn thuật toán giải phương trình $ax+b=0$ bằng mã giả

1. Nhập a, b

2. if(a=0)

if(b=0) xuất “Phương trình vô số nghiệm”

else xuất “Phương trình vô nghiệm”

else

xuất “Phương trình có nghiệm $x = \frac{-b}{a}$ ”

Ví dụ 5: Trình bày **thuật toán** tìm phần tử lớn nhất trong ba số a, b, c bằng lưu đồ và mã giả.

Nhập: a, b, c

Xuất: max là giá trị lớn nhất trong ba số a, b, c

III. CÁC TÍNH CHẤT CỦA THUẬT TOÁN

Để có thể lập trình giải bài toán trên máy tính, ta cần có một thuật toán bảo đảm những tính chất nhất định:

Tính xác định (*definiteness*): Các bước trong thuật toán phải chính xác.

Tính hữu hạn (*finiteness*): Thuật toán phải cho ra lời giải (hay kết quả) sau một số hữu hạn bước.

Tính đúng: Thuật toán phải cho kết quả như mong muốn.

Tính hiệu quả về thời gian: Thuật toán cần phải được thực hiện một cách chính xác và trong một khoảng thời gian cho phép.

Tính tổng quát: Thuật toán phải áp dụng được cho tất cả các bài toán có dạng như mong muốn, chứ không phải chỉ áp dụng được cho một số trường hợp đặc biệt nào đó.

Trong các tính chất trên, 3 tính chất cơ bản của thuật toán đòi hỏi phải được thỏa mãn là tính xác định, tính hữu hạn và tính đúng.

Ví dụ 6 : Trình bày **thuật toán** tìm phần tử lớn nhất trong một dãy hữu hạn $a_1, a_2, \dots, a_n$ các số nguyên bằng mã giả.

Nhập: dãy số $a_1, a_2, \dots, a_n$

Xuất: max là giá trị lớn nhất trong dãy số $a_1, a_2, \dots, a_n$

Thuật toán bằng mã giả:

1. Nhập dãy số $a_1, a_2, \dots, a_n$

2. $\text{max} = a_1$

3. for ($i = 2; i \leq n; i++$)

 if $\text{max} < a_i$ $\text{max} = a_i$

4. Xuất: max là giá trị lớn nhất trong dãy số.