

2.1. Định nghĩa thuật toán

- Là một khái niệm cơ sở của toán học và tin học.
- Bao gồm một dãy hữu hạn các lệnh/chỉ thị rõ ràng và có thể thi hành được để hướng dẫn thực hiện một hành động nhằm đạt được mục tiêu đề ra.
- Thuật toán là sự thể hiện của một phương pháp để giải quyết một vấn đề.

Nội dung chương này

- 2.1. Định nghĩa thuật toán
- 2.2. Biểu diễn thuật toán
- 2.3. Một số thuật toán thông dụng
- 2.4. Thuật toán đệ quy
- 2.5. Thuật giải heuristic

Ví dụ 1: Thuật toán tìm phần tử lớn nhất của một dãy hữu hạn các số nguyên

- Các bước:
 - 1. Đặt giá trị lớn nhất tạm thời là số nguyên đầu tiên.
 - 2. So sánh số nguyên kế tiếp trong dãy với giá trị lớn nhất tạm thời, nếu số nguyên này lớn hơn giá trị lớn nhất tạm thời thì đặt giá trị lớn nhất tạm thời bằng số nguyên này.
 - 3. Lặp lại bước 2 nếu còn số nguyên trong dãy chưa được xét.
 - 4. Dừng nếu không còn số nguyên nào trong dãy chưa được xét. Giá trị lớn nhất tạm thời lúc này chính là giá trị lớn nhất trong dãy số.

Ví dụ 2: Thuật toán giải phương trình bậc

hai: $ax^2 + bx + c = 0$ ($a \neq 0$)

- 1. Nhập 3 hệ số a, b, c
- 2. Tính giá trị $\Delta = b^2 - 4*a*c$
- 3. Xét dấu Δ . Nếu $\Delta > 0$ thì thực hiện các thao tác sau đây:
 - 3.1. Tính các nghiệm theo các công thức:
 - $x_1 = (-b - \sqrt{\Delta}) / (2*a)$
 - $x_2 = (-b + \sqrt{\Delta}) / (2*a)$
 - 3.2. Xuất kết quả: phương trình có hai nghiệm x_1 và x_2 .
- 4. Nếu Δ là 0 thì xuất kết quả: phương trình có nghiệm kép là $-b / (2*a)$
- 5. Nếu $\Delta < 0$ thì xuất kết quả: phương trình vô nghiệm
- 6. Dừng thuật toán

5

Các đặc trưng của thuật toán

- Nhập (input):** có các giá trị nhập từ một tập hợp nhất định.
- Xuất (output):** từ mỗi giá trị của tập hợp nhập, tạo ra giá trị xuất thuộc một tập hợp nhất định.
- Tính xác định (definiteness):** các bước chính xác, rõ ràng.
- Tính hữu hạn (finiteness):** cho ra kết quả sau một số hữu hạn bước.
- Tính hiệu quả (effectiveness):** được đánh giá dựa trên một số tiêu chuẩn (khối lượng tính toán, không gian, thời gian sử dụng).
- Tính tổng quát (generalness):** áp dụng được cho tất cả các bài toán có dạng như mong muốn

6

2.2. Biểu diễn thuật toán

- Sử dụng các ngôn ngữ:
 - Ngôn ngữ tự nhiên
 - Ngôn ngữ lưu đồ (sơ đồ khối)
 - Ngôn ngữ tựa ngôn ngữ lập trình (mã giả)
 - Ngôn ngữ lập trình

7

Ngôn ngữ lưu đồ

- Các thành phần:
 - Nút giới hạn: được biểu diễn bởi hình ôvan có ghi chữ bên trong, gồm có nút đầu và nút cuối:

BẮT ĐẦU

KẾT THÚC

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

tăng k

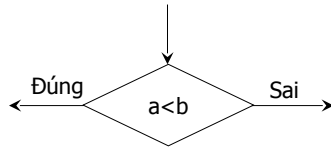
- Nút nhập/xuất dữ liệu:

Đọc a và
b

8

Ngôn ngữ lưu đồ (2)

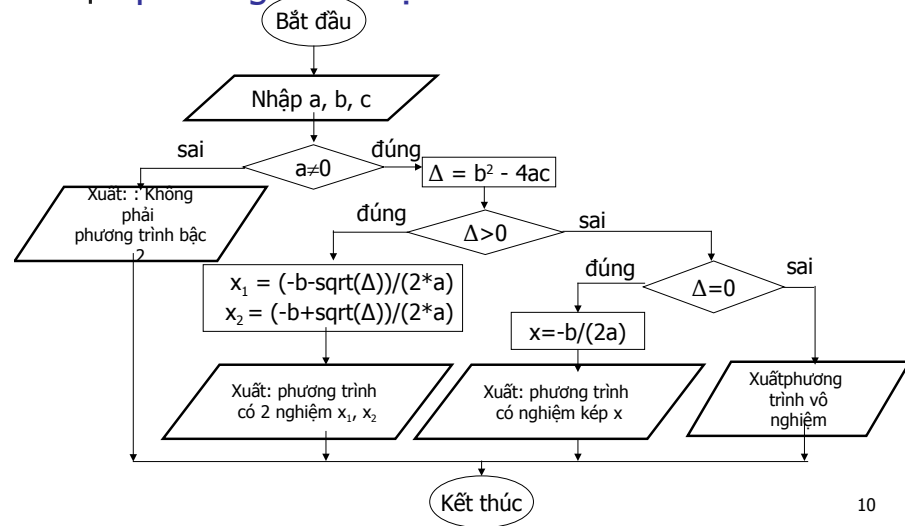
- Nút điều kiện: là một hình thoi có ghi điều kiện cần kiểm tra, thường có 1 cung đi vào và 2 cung đi ra (tương ứng với 2 trường hợp đúng/sai)



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

9

Ví dụ: lưu đồ biểu diễn thuật toán giải phương trình bậc 2



10

Mã giả

- Sử dụng mệnh đề có cấu trúc chuẩn hóa và vẫn dùng ngôn ngữ tự nhiên.
- Sử dụng các ký hiệu toán học, các biến, cấu trúc kiểu thủ tục.
- Hành động gán:
 - $i \leftarrow i+1$
- Tiện lợi, đơn giản, vẫn dễ hiểu.

11

Mã giả (2)

- Các cấu trúc thường gặp:
 - Cấu trúc chọn:
 - if (điều kiện) then (hành động) end if
 - if (điều kiện) then (hành động 1) else (hành động 2) end if
 - Cấu trúc lặp
 - while (điều kiện) do (hành động) end while
 - repeat (hành động) until (điều kiện)
 - for (biến)=(giá trị đầu) to (giá trị cuối) do (hành động) end for
 - for (biến)=(giá trị cuối) downto (giá trị đầu) do (hành động) end for
 - Cấu trúc nhảy
 - goto nhãn x;

12

Ví dụ: thuật toán giải phương trình bậc 2

- **Nhập:** các hệ số a, b, c
- **Xuất:** kết luận về nghiệm của phương trình bậc hai
- **Thuật toán:**
- if a = 0 then
- Xuất: Không phải phương trình bậc hai, Dừng
- end if
- delta $\leftarrow b^2 - 4*a*c$
- if delta > 0 then
- x1 $\leftarrow (-b - \sqrt{\Delta}) / (2*a)$
- x2 $\leftarrow (-b + \sqrt{\Delta}) / (2*a)$
- Xuất: x1 và x2, Dừng
- else if delta = 0 then x1 $\leftarrow -b / (2*a)$, Xuất: nghiệm kép x12
- else Xuất: phương trình vô nghiệm
- end if

13

2.3. Một số thuật toán thông dụng

- Thuật toán kiểm tra số nguyên tố
- Thuật toán tìm USCLN, BSCNN của 2 số nguyên
- Thuật toán tìm phần tử lớn nhất trong một dãy
- Thuật toán sắp xếp
- Thuật toán tìm kiếm

14

Tìm phần tử lớn nhất trong một dãy hữu hạn số

- **Nhập:** dãy số a[1], a[2], a[3],... a[n]
- **Xuất:** max là giá trị lớn nhất trong dãy số đã cho

Thuật toán:

max $\leftarrow a[1]$

for i = 2 to n do

 if max < a[i] then

 max $\leftarrow a[i]$

 end if

end for

Xuất: max là giá trị lớn nhất trong dãy số

15

2.4. Thuật toán đệ quy

- Có một số trường hợp, cách giải có thể vi phạm các tính chất của thuật toán nhưng lại khá đơn giản và được chấp nhận.
- Bài toán có thể được phân tích và đưa tới việc giải một bài toán cùng loại nhưng cấp độ thấp hơn.
- Ví dụ:
 - Định nghĩa giai thừa
 - $0! = 1$
 - $n! = n*(n-1)!$ với $n > 0$
 - Định nghĩa dãy số Fibonacci: 1, 1, 2, 3, 5, 8, 13, ...
 - $f_1 = 1,$
 - $f_2 = 1,$
 - $f_n = f_{n-1} + f_{n-2}$

16

Thuật toán đệ quy (2)

- Thuật toán đệ quy tính giai thừa của 1 số tự nhiên:
 - Input: số tự nhiên n
 - Output: $F(n)$ bằng $n!$
 - Thuật giải:
 - $F \leftarrow 1$
 - if $n > 0$ then $F \leftarrow F(n-1) * n$
 - Output F

17

Thuật toán đệ quy (3)

- Thuật toán đệ quy tính số hạng thứ n của dãy số Fibonacci:
 - Input: số tự nhiên n
 - Output: $F(n)$ bằng số hạng thứ n của dãy
 - Thuật giải:
 - if $n=1$ or $n=2$ then $F \leftarrow 1$
 - if $n > 2$ then $F \leftarrow F(n-1) + F(n-2)$
 - Output F

18

Thuật toán đệ quy (4)

- Đặc điểm của thuật toán đệ quy:
 - Có 1 trường hợp cơ sở/trường hợp dừng
 - Có phần đệ quy bên trong thuật toán (nó gọi đến chính nó)
 - Có sự biến đổi tiến tới trường hợp cơ sở.

19

Bài tập

- Viết thuật toán tìm USCLN của hai số tự nhiên
- Viết thuật toán tìm BSCNN của hai số tự nhiên
- Viết thuật toán tìm phần tử lớn nhất trong một dãy số hữu hạn
- Viết thuật toán sắp xếp
- Viết thuật toán tìm kiếm

20

2.5. Thuật giải heuristic

- Thường tìm được lời giải tốt (những chưa chắc đã tốt nhất)
- Dễ dàng và nhanh chóng hơn so với giải thuật tối ưu
- Thể hiện một cách hành động khá tự nhiên, gần gũi với suy nghĩ và hành động của con người.

21

Thuật giải heuristic (2)

- Các nguyên lý
 - Nguyên lý vét cạn thông minh: trong bài toán tìm kiếm khi không gian tìm kiếm lớn => giới hạn không gian tìm kiếm hoặc thực hiện dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu
 - Nguyên lý tham lam: lấy tiêu chuẩn tối ưu toàn cục làm tiêu chuẩn chọn lựa hành động cục bộ của từng bước trong quá trình tìm kiếm lời giải
 - Nguyên lý thứ tự: thực hiện hành động theo thứ tự hợp lý của không gian khảo sát nhằm nhanh chóng đạt được một lời giải tốt.

22