

TIN HỌC ĐẠI CƯƠNG

Bài 3. Giải quyết bài toán

Bùi Trọng Tùng, SoICT, HUST

Nội dung

1. Thuật toán
2. Biểu diễn thuật toán
3. Một số thuật toán cơ bản

Nội dung

- ⇒ 1. Thuật toán
- 2. Biểu diễn thuật toán
- 3. Một số thuật toán cơ bản

3

Khái niệm Bài toán (problem)

- “Bài toán” hay “Vấn đề”
 - Vấn đề có nghĩa rộng hơn bài toán
 - Bài toán là một loại vấn đề mà để giải quyết phải liên quan ít nhiều đến tính toán: bài toán trong vật lý, hóa học, xây dựng, kinh tế...
- Biểu diễn vấn đề-bài toán: $A \rightarrow B$
 - A: Giả thiết, điều kiện ban đầu
 - B: Kết luận, mục tiêu cần đạt
- Giải quyết vấn đề-bài toán
 - Từ A dùng một số hữu hạn các bước suy luận có lý hoặc hành động thích hợp để đạt được B
 - Trong Tin học, A là đầu vào, B là đầu ra

4

Giải quyết bài toán bằng máy tính

- Máy tính không thể dùng để giải quyết các vấn đề liên quan đến hành động vật lý hoặc biểu thị cảm xúc
- Máy tính chỉ làm được những gì mà nó được bảo phải làm. Máy tính không thông minh, nó không thể tự phân tích vấn đề và đưa ra giải pháp.
- Lập trình viên là người phân tích vấn đề, tạo ra các chỉ dẫn để giải quyết vấn đề (chương trình), và máy tính sẽ thực hiện các chỉ dẫn đó

5

Giải quyết bài toán bằng máy tính

- Phương án giải quyết bài toán được gọi là thuật toán/giải thuật trong tính toán
- Một thuật toán là:
 - một dãy **hữu hạn** các thao tác và **trình tự** thực hiện các thao tác đó sao cho sau khi thực hiện dãy thao tác này theo trình tự đã chỉ ra, với **đầu vào** (input) ta thu được kết quả **đầu ra** (output) mong muốn.

6

Các đặc trưng của thuật toán

- Tính dừng: phải cho ra kết quả sau một số hữu hạn các bước
- Tính đúng đắn: kết quả tính toán của giải thuật là chính xác
- Tính xác định: các bước thực hiện có trình tự xác định
- Tính khách quan: cho kết quả như nhau khi chạy trên các máy tính khác nhau
- Tính tổng quát: áp dụng cho các bài toán cùng dạng mong muốn
- Tính hiệu quả: đánh giá dựa trên khối lượng các phép toán cần thực hiện

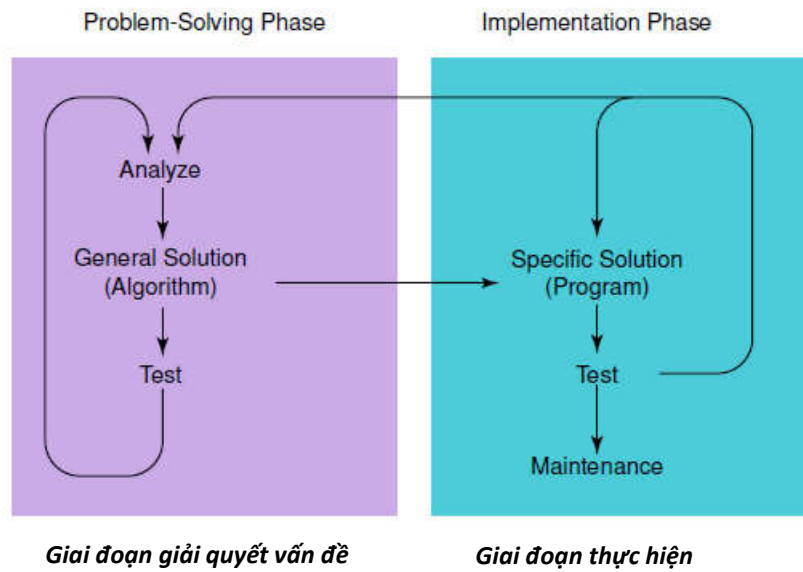
7

Giải quyết bài toán bằng máy tính

- Không chỉ đơn giản là lập trình
- Phức tạp, gồm nhiều giai đoạn phát triển
- Các giai đoạn quan trọng
 - Bước 1. Xác định yêu cầu bài toán
 - Bước 2. Phân tích và thiết kế bài toán
 - Lựa chọn phương án giải quyết (thuật toán)
 - Xây dựng thuật toán
 - Bước 3. Lập trình
 - Bước 4. Kiểm thử và hiệu chỉnh chương trình
 - Bước 5. Triển khai và bảo trì

8

Hai giai đoạn chính để hiện thực hóa bài toán



Nội dung

1. Thuật toán
- ⇒ 2. Biểu diễn thuật toán
3. Một số thuật toán cơ bản

Biểu diễn thuật toán

- **Cách 1:** Ngôn ngữ tự nhiên
- **Cách 2:** Ngôn ngữ lưu đồ (lưu đồ/sơ đồ khối)
- **Cách 3:** Mã giả (pseudocode) gọi là ngôn ngữ mô phỏng chương trình PDL (Programming Description Language).
- **Cách 4:** Các ngôn ngữ lập trình như Pascal, C/C++ hay Java. Tuy nhiên, không nhất thiết phải sử dụng đúng ký pháp của các ngôn ngữ đó mà có thể được bỏ một số ràng buộc.

11

Nội dung

1. Thuật toán

→ 2. Biểu diễn thuật toán

2.1. Ngôn ngữ tự nhiên

2.2. Lưu đồ thuật toán

3. Một số thuật toán cơ bản

12

Nội dung

1. Thuật toán
2. Biểu diễn thuật toán
 - 2.1. Ngôn ngữ tự nhiên
 - 2.2. Lưu đồ thuật toán
3. Một số thuật toán cơ bản

13

2.1. Ngôn ngữ tự nhiên

- Sử dụng một loại ngôn ngữ tự nhiên để liệt kê các bước của thuật toán
- Ưu điểm
 - Đơn giản
 - Không yêu cầu người viết và người đọc phải có kiến thức nền tảng
- Nhược điểm
 - Dài dòng
 - Không làm nổi bật cấu trúc của thuật toán
 - Khó biểu diễn với những bài toán phức tạp

14

Ví dụ

- Ví dụ 1: Tính tổng, tích, hiệu, thương của hai số a và b.
 - Đầu vào: Hai số a và b
 - Đầu ra: Tổng, tích, hiệu và thương của a và b.
- Ý tưởng:
 - Tính tổng, tích, hiệu của a và b
 - Tính thương a/b
 - Nếu b khác 0, tính thương
 - Nếu b bằng 0, đưa ra thông báo không thực hiện được phép chia

15

Mô tả bằng ngôn ngữ tự nhiên

- B1: Nhập số a và số b.
- B2: tong $\leftarrow a + b$; hieu $\leftarrow a - b$; tich $\leftarrow a * b$
- B3: Hiển thị tong, hieu, tich
- B3: Nếu $b = 0$, thông báo lỗi “Không thực hiện được phép chia”.
Ngược lại thương $\leftarrow a/b$. Hiển thị thương
- B5: Kết thúc

16

Nội dung

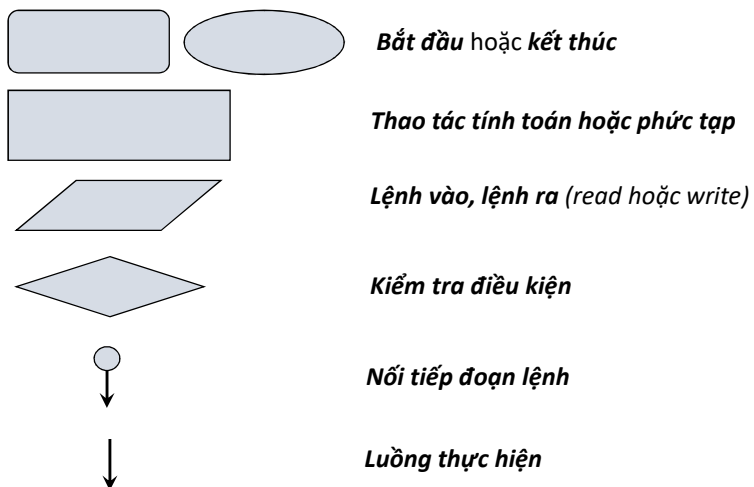
1. Thuật toán
2. Biểu diễn thuật toán
 - 2.1. Ngôn ngữ tự nhiên
 - 2.2. Lưu đồ thuật toán
3. Một số thuật toán cơ bản



17

2.2. Lưu đồ thuật toán

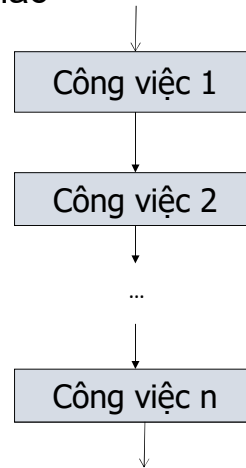
Một số khối trong sơ đồ khối dùng biểu diễn thuật toán



18

Cấu trúc tuần tự

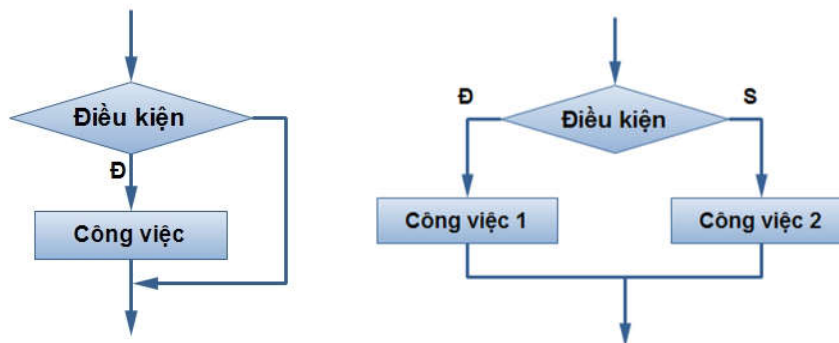
- Các bước được thực hiện theo 1 trình tự tuyến tính, hết bước này đến bước khác



19

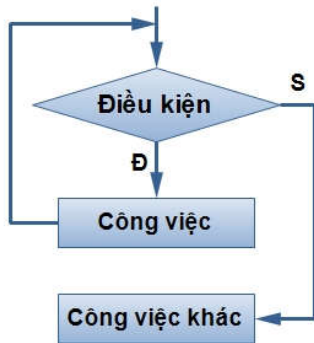
Cấu trúc rẽ nhánh

- Nếu biểu thức điều kiện đúng (giá trị chân lý là True) thực hiện công việc 1.
- Nếu biểu thức điều kiện sai (giá trị chân lý là False) thực hiện công việc 2.

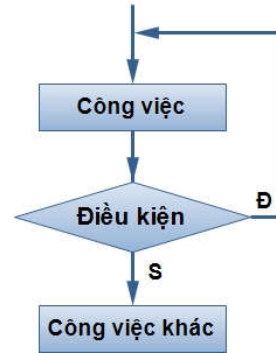


20

Cấu trúc lặp



- Khi biểu thức điều kiện còn đúng, thực hiện công việc

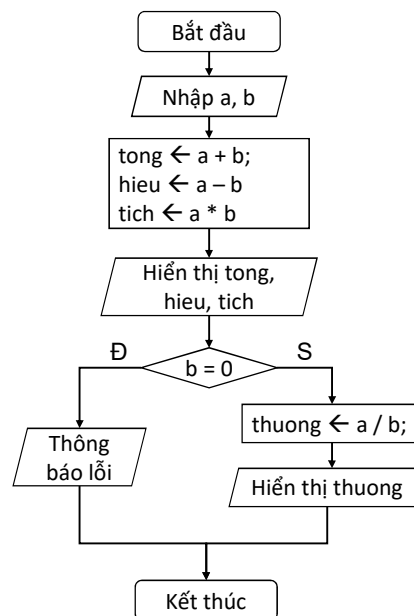


- Thực hiện công việc khi biểu thức điều kiện còn đúng

21

Ví dụ

- B1: Nhập số a và số b.
- B2: $tong \leftarrow a + b$;
 $hieu \leftarrow a - b$
 $tich \leftarrow a * b$
- B3: Hiển thị tong, hieu, tich
- B4: Nếu $b = 0$, thông báo lỗi “Không thực hiện được phép chia”.
Ngược lại $thuong \leftarrow a / b$.
Hiển thị thuong
- B5: Kết thúc



22

Lưu đồ thuật toán

- Ưu điểm
 - Trực quan, dễ hiểu, dễ thiết kế
 - Cung cấp toàn cảnh, tổng quan về thuật toán
- Nhược điểm
 - Cồng kềnh, đặc biệt với bài toán phức tạp

23

Nội dung

1. Thuật toán
2. Biểu diễn thuật toán
- ⇒ 3. Một số thuật toán cơ bản
 - 3.1. Thuật toán số học
 - 3.2. Thuật toán về dãy
 - 3.3. Thuật toán đệ quy

24

Nội dung

1. Thuật toán
2. Biểu diễn thuật toán
3. Một số thuật toán cơ bản

- ⇒
- 3.1. Thuật toán số học
 - 3.2. Thuật toán về dãy
 - 3.3. Thuật toán đệ quy

25

Giải phương trình bậc 1

- Bài toán: Giải phương trình bậc I
 - Đầu vào: Hai hệ số a, b
 - Đầu ra: Nghiệm của phương trình $ax + b = 0$
- Ý tưởng:
 - Lần lượt xét $a = 0$ rồi xét $b = 0$ để xét các trường hợp của phương trình

26

Mô tả tuần tự các bước

- B1: Nhập a và b.
 - B2: Nếu $a \neq 0$ thì $x \leftarrow -b/a$. Hiển thị “Phương trình có 1 nghiệm duy nhất x”.
 Ngược lại sang B3
 - B3: Nếu $b \neq 0$ thì hiển thị “Phương trình vô nghiệm”.
 Ngược lại Hiển thị “Phương trình vô số nghiệm”
- B4: Kết thúc

27

Lưu đồ thuật toán

28

Bài tập

- Bài toán: Giải phương trình bậc II
 - Đầu vào: Ba hệ số a, b, c
 - Đầu ra: Nghiệm của phương trình
$$ax^2 + bx + c = 0 \quad (a \neq 0)$$
- Ý tưởng:
 - Tính định thức và biện luận nghiệm

29

Bài toán số nguyên tố

- Cho một số nguyên dương p . Làm thế nào để biết được p có phải số nguyên tố hay không?
 - Input: p nguyên dương
 - Output: kết luận về tính nguyên tố của p
- Ý tưởng?
 - Nếu $p = 0$ hoặc 1 : Không phải số nguyên tố
 - Nếu $p > 1$:
 - Kiểm tra từ 2 đến $p - 1$ có phải là ước số của p không
 - Nếu có thì kết luận p không là số nguyên tố, ngược lại không có số nào thì kết luận p là số nguyên tố

30

Mô tả các bước

31

Lưu đồ thuật toán

32

Tìm UCLN, BCNN của hai số

- B1 : Nhập 2 số a, b
- B2 : $x \leftarrow a, y \leftarrow b$
- B3 : Nếu $y \neq 0$, sang B4
 Ngược lại, sang bước 5
- B4 : $r \leftarrow x \bmod y$
 $x \leftarrow y$
 $y \leftarrow r$
 Quay lại B3.
- B5: $ucln \leftarrow x, bcnn \leftarrow (a*b) / ucln$.
- B6: Hiển thị ucln, bcnn.
- B7: Kết thúc

33

Tìm ước số

Tìm tất cả các ước số của một số nguyên dương N

- Ý tưởng : Duyệt lần lượt các giá trị từ 1 tới N và in ra nếu giá trị đó là ước số của N.

Giải thuật

- B1 : Nhập N
- B2 : Gán $i \leftarrow 1$
- B3 : Nếu $i \leq N$ tới B4
 Ngược lại kết thúc
- B4 : Nếu N chia hết cho i, hiển thị i.
- B5 : $i \leftarrow i + 1$. Quay lại B3

34

Tìm ước số chẵn

Tìm tất cả các ước số chẵn của một số nguyên dương N

- Ý tưởng : Duyệt lần lượt các số chẵn từ 2 tới N và in ra nếu giá trị đó là ước số của N.

Giải thuật

- B1 : Nhập N
- B2 : Gán $i \leftarrow 2$
- B3 : Nếu $i \leq N$ tới B4
 Ngược lại, kết thúc
- B4 : Nếu N chia hết cho i, xuất i.
- B5 : $i \leftarrow i + 2$. Quay lại B3
- B6 : Kết thúc

35

Số hoàn hảo

Kiểm tra tính hoàn hảo của số nguyên dương N.

- B1 : Nhập N
- B2 : Gán $i \leftarrow 1$, tongUoc $\leftarrow 0$,
- B3 : Nếu $i < N$ tới B4
 Ngược lại, tới B6
- B4 : Nếu N chia hết cho i, gán tongUoc \leftarrow tongUoc + i
- B5 : $i \leftarrow i + 1$. Quay lại B3
- B6 : Nếu tongUoc = N, kết luận N là số hoàn hảo.
 Ngược lại, kết luận N không phải là số hoàn hảo
- B7 : Kết thúc

36

Tính số Fibonacci

Tìm số Fibonacci thứ N

Thuật giải

- B1 : Nhập N
- B2 : Gán $i \leftarrow 2$, $F_{n-1} \leftarrow 1$, $F_{n-2} \leftarrow 1$, $F_n \leftarrow 1$
- B3 : Nếu $i < N$ tới B4.
 Ngược lại, tới B5
- B4 : Gán $F_n \leftarrow F_{n-1} + F_{n-2}$
 Gán $F_{n-2} \leftarrow F_{n-1}$
 Gán $F_{n-1} \leftarrow F_n$
 Gán $i \leftarrow i + 1$
 Quay lại B3
- B5: Hiện thị F_n .
- B6: Kết thúc.

37

Nội dung

1. Thuật toán
2. Biểu diễn thuật toán
3. Một số thuật toán cơ bản
 - 3.1. Thuật toán số học
 - ⇒ 3.2. Thuật toán về dãy
 - 3.3. Thuật toán đệ quy

38

3.2. Thuật toán về dãy

- Làm việc với một dãy số
- Các bài toán điển hình
 - Tìm số lớn nhất, nhỏ nhất trong dãy
 - Kiểm tra dãy có phải là dãy tăng hoặc dãy giảm
 - Sắp xếp dãy tăng dần hoặc giảm dần
 - Tìm trong dãy có phần tử nào bằng một giá trị cho trước
 - Tính trung bình cộng của dãy
 - ...

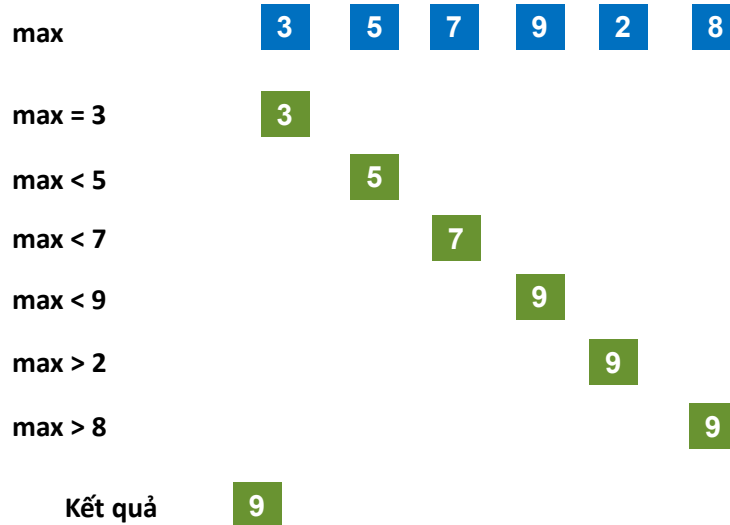
39

Tìm số lớn nhất trong dãy

- Bài toán: Tìm giá trị lớn nhất của một dãy số nguyên có n số
 - Đầu vào: Số nguyên dương n và n số nguyên a_0, a_1, \dots, a_{n-1}
 - Đầu ra: số nguyên lớn nhất của dãy
- Ý tưởng:
 - Khởi tạo giá trị $\text{max} = a_0$
 - Lần lượt so sánh max với a_i với $i = 1, 2, \dots, n-1$
nếu $\text{max} < a_i$ ta gán giá trị mới cho $\text{max} = a_i$

40

Ý tưởng



41

Mô tả tuần tự các bước

- B1: Nhập n và dãy số a_0, a_1, \dots, a_{n-1} .
- B2: $\max \leftarrow a_0$;
 $i = 1$ (*chỉ số của phần tử tiếp theo*)
- B3: Nếu $i \leq n-1$, sang bước 4
Ngược lại in ra giá trị \max . Kết thúc.
- B4: Nếu $\max < a_i$, $\max \leftarrow a_i$
- B5: $i \leftarrow i + 1$ (*chỉ số của phần tử kết tiếp*).
Quay lại B3.
- B6: Kết thúc.

42

Lưu đồ thuật toán

43

Thuật toán sắp xếp lựa chọn

- Bài toán: Sắp xếp lựa chọn (Selection Sorting)
 - Đầu vào: Dãy A gồm n số nguyên a_0, a_1, \dots, a_{n-1}
 - Đầu ra: Dãy A được sắp lại theo thứ tự tăng.
- Ý tưởng:
 - Nhận xét: Dãy có 2 đoạn:
 - Đoạn đầu: gồm các phần tử đã được sắp xếp đúng thứ tự tăng. Ban đầu đoạn này chưa có phần tử nào
 - Đoạn sau: gồm các phần tử chưa được sắp xếp. Ban đầu đoạn này có n phần tử a_0, a_1, \dots, a_{n-1}
 - So sánh phần tử đầu tiên trong đoạn chưa được sắp xếp với các phần tử sau nó một cách lần lượt. Nếu chưa đúng thứ tự thì đổi chỗ. Sau mỗi lượt phần tử đầu tiên của đoạn này trở thành đúng thứ tự.

44

Thuật toán sắp xếp lựa chọn – Ý tưởng

- Lần sắp xếp thứ 1:
 - Đoạn đã được sắp xếp: chưa có phần tử nào
 - Đoạn chưa được sắp xếp: có n phần tử a_0, a_1, \dots, a_{n-1}
 - So sánh a_0 với a_j ($1 \leq j \leq n-1$). Nếu $a_0 > a_j$ thì đổi chỗ
 - Sau lượt sắp xếp này a_0 đã đúng thứ tự
- Lần sắp xếp thứ 2:
 - Đoạn đã được sắp xếp: a_0
 - Đoạn chưa được sắp xếp: a_1, a_2, \dots, a_{n-1}
 - So sánh a_1 với a_j ($2 \leq j \leq n-1$). Nếu $a_1 > a_j$ thì đổi chỗ
 - Sau lượt sắp xếp này a_1 đã đúng thứ tự

45

Thuật toán sắp xếp lựa chọn – Ý tưởng

- Lần sắp xếp thứ i :
 - Đoạn đã được sắp xếp: a_0, a_1, \dots, a_{i-2}
 - Đoạn chưa được sắp xếp: $a_{i-1}, a_i, \dots, a_{n-1}$
 - So sánh a_{i-1} với a_j ($i \leq j \leq n-1$). Nếu $a_{i-1} > a_j$ thì đổi chỗ
 - Sau lượt sắp xếp này a_{i-1} đã đúng thứ tự
- Thuật toán dừng khi nào?

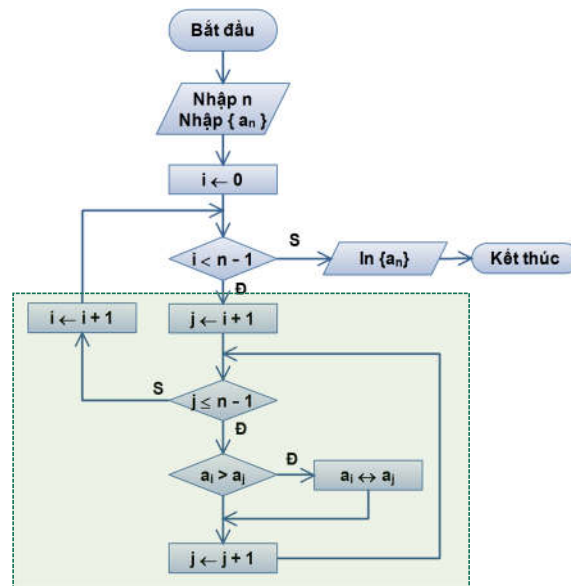
46

Mô tả tuần tự các bước

- B1: Nhập số n và dãy số a_0, a_1, \dots, a_{n-1}
- B2: $i \leftarrow 0$ (phần tử đầu tiên của dãy chưa được sắp xếp)
- B3: Nếu $i < n - 1$ tới B4
Ngược lại, dãy đã được sắp xếp xong. Kết thúc.
- B4: Gán $j \leftarrow i + 1$ (chỉ số của phần tử sau trong đoạn chưa được sắp xếp)
- B5: Nếu $j \leq n - 1$ (chỉ số cuối của dãy chưa được sắp xếp) tới B6
- B6: Nếu $a_i > a_j$ thì đổi chỗ hai phần tử.
- B7: Gán $j \leftarrow j + 1$. Quay lại B5.

47

Lưu đồ thuật toán



48

Nội dung

1. Thuật toán
2. Biểu diễn thuật toán
3. Một số thuật toán cơ bản
 - 3.1. Thuật toán số học
 - 3.2. Thuật toán về dãy
 - 3.3. Thuật toán đệ quy

49

3.3. Thuật toán đệ quy

- Với bài toán có thể được phân tích và đưa tới việc giải một bài toán cùng loại nhưng cấp độ thấp hơn
 - độ lớn dữ liệu nhập nhỏ hơn
 - giá trị cần tính toán nhỏ hơn
- Tự thực hiện lại thuật toán
- Ví dụ:
 - Giai thừa: $n! = (n-1)! * n$
 - Dãy số Fibonacci: 0, 1, 1, 2, 3, 5, 8...
 - $F(n) = F(n-1) + F(n-2)$

50

Thuật toán đệ quy

- Để xây dựng thuật toán đệ quy, cần xác định:
 - Trường hợp cơ bản: (Các) trường hợp không cần thực hiện lại thuật toán.
 - Phân tổng quát: Có yêu cầu gọi đệ quy
 - Cần xác định nguyên lý đưa trường hợp tổng quát về trường hợp cơ bản
 - Đảm bảo tính dừng của giải thuật đệ quy - chắc chắn từ trường hợp tổng quát sẽ đến được trường hợp cơ bản

51

Ví dụ

- Tính giai thừa của n:
 - Trường hợp cơ bản: $0! = 1$
 - Trường hợp tổng quát: $n! = (n-1)! * n$
- Xây dựng dãy Fibonacci
 - Trường hợp cơ bản: $F(0) = F(1) = 1$
 - Trường hợp tổng quát: $F(n) = F(n-1) + F(n-2)$

52

Tính giai thừa - Thuật toán đệ quy

- Input: số tự nhiên n
- Output: $GT(n)=n!$
- Thuật giải:

Nhập n

$GT := 1;$

Nếu $n > 0$ **thì**

$GT := GT(n-1) * n;$

Xuất GT

53

Bài tập

- Xây dựng thuật toán cho bài toán tìm số Fibonacci $F(n)$

54