



Cấu trúc dữ liệu và giải thuật

Trần Hồng Diệp

E-mail: diepthd@gmail.com

Tài liệu

- [1]: Đỗ Xuân Lôi “Cấu trúc dữ liệu và giải thuật”, NXB Khoa học Kỹ thuật, 1996 / 2005.
- [2]: Nguyễn Đức Nghĩa, Nguyễn Tô Thành “Toán rời rạc”, NXB Đại học Quốc gia Hà nội, 2005.
- [3]: Nguyễn Đức Nghĩa “Algorithmics-Thiết kế và phân tích thuật toán”, Tài liệu của Trường Đại học Bách khoa Hà Nội.



Nội dung chương trình



Chương 1: Tổng quan.

Chương 2: Thiết kế và phân tích giải thuật.

Chương 3: Giải thuật đệ qui.

Chương 4: Array và List.

Chương 5: Stack và Queue.

Chương 6: Tree.

Chương 7: Sắp xếp.

Chương 8: Tìm kiếm

CHƯƠNG 1: Tổng quan

- I. Mỗi quan hệ Giải thuật – Cấu trúc Dữ liệu.
- II. Cấu trúc dữ liệu – một số vấn đề liên quan.
- III. Ngôn ngữ biểu đạt giải thuật.

I. Mỗi quan hệ Giải thuật & Cấu trúc Dữ liệu.

❖ Kiến thức cũ:

◆ Giải thuật – Algorithms?

- ✓ Tập các chỉ thị - Statements
- ✓ Hữu hạn
- ✓ Thứ tự.
- ✓ Đối tượng tác động
- ✓ Mục đích

I. Mối quan hệ Giải thuật & Cấu trúc Dữ liệu.

❖ Kiến thức cũ:

◆ Dữ liệu có cấu trúc – Data Structures?

- ✓ Mảng – Array?
- ✓ Bản ghi – Record?
- ✓

⇒ Là đối tượng tác động của giải thuật.

⇒ Được cấu trúc từ các kiểu dữ liệu khác.

I. Mỗi quan hệ Giải thuật & Cấu trúc Dữ liệu.

❖ Vài ví dụ:

- ◆ Chương trình tính trung bình cộng của một dãy các số nguyên.
- ◆ Chương trình lưu trữ, sắp xếp, tìm kiếm các hồ sơ sinh viên.

I. Mỗi quan hệ Giải thuật & Cấu trúc Dữ liệu.

❖ Vấn đề đặt ra:

- ◆ Giải thuật phải xác định là tác động trên kiểu dữ liệu như thế nào.
- ◆ Dữ liệu phải phù hợp với giải thuật đưa ra.
- ◆ Cấu trúc dữ liệu hợp lý có khả năng viết được một giải thuật tốt, đơn giản.

II. Cấu trúc dữ liệu & một số vấn đề liên quan

- ◆ Dữ liệu nguyên tử - Atoms của mỗi cấu trúc dữ liệu.
- ◆ Cấu trúc tiền định của các ngôn ngữ bậc cao – Predefined Data Structures.
- ◆ Phép xử lý tương ứng trên cấu trúc dữ liệu.
- ◆ Phân biệt hai khái niệm *Cấu trúc dữ liệu* – Data Structures & *Cấu trúc lưu trữ* - Storage Structures.

III. Ngôn ngữ biểu diễn giải thuật

❖ Vấn đề:

- ◆ Cần có một ngôn ngữ để diễn đạt các cách cấu trúc và các phép xử lý trên chúng một cách rõ ràng, dễ hiểu và ngắn gọn.
- ◆ Nhưng cũng không cần một ngôn ngữ bậc cao cụ thể:
 - ✓ Nội dung chương trình không chủ yếu quan tâm đến các ngôn ngữ bậc cao cụ thể.
 - ✓ Các ngôn ngữ bậc cao luôn có các yêu cầu về cú pháp riêng làm cho việc trình bày vấn đề trở nên gò bó nặng nề.

III. Ngôn ngữ biểu diễn giải thuật

3.1. Cấu trúc chương trình:

```
Program <Tên chương trình>;  
  Type <Khai báo cấu trúc>;  
  Const <Khai báo hằng>;  
  Var <Khai báo biến>;  
  Procedure <Khai báo thủ tục>;  
  Function <Khai báo hàm>;  
Begin  
  <Thân chương trình>  
End.
```

III. Ngôn ngữ biểu diễn giải thuật

3.1. Cấu trúc chương trình:

- ♦ Mỗi chương trình được đặt một tên riêng để phân biệt, không hạn chế kí tự, được viết liên tục không có các kí tự trống và nên viết in hoa (có thể viết chữ Việt).
- ♦ Lệnh được kết thúc bởi kí tự chấm phẩy “;”
- ♦ Cấu trúc dữ liệu mới trong khuôn khổ cấu trúc tiền định được định nghĩa sau từ khoá *Type*.

III. Ngôn ngữ biểu diễn giải thuật

3.1. Cấu trúc chương trình:

- ♦ Thân chương trình được bao trong cặp từ khoá:

{ <Thân chương trình> }

- ♦ Lời chú thích được viết cuối dòng lệnh hoặc trên một dòng riêng và được bao trong cặp ngoặc (có thể viết chữ Việt):

/ <Lời chú thích> */*

III. Ngôn ngữ biểu diễn giải thuật

3.2. Chương trình con:

- ♦ Định nghĩa chương trình con dạng thủ tục:

Procedure <Tên thủ tục>(<Danh sách tham số>);

Type <Cấu trúc cục bộ>

Var <Biến cục bộ>;

{

<Thân thủ tục>

}

III. Ngôn ngữ biểu diễn giải thuật

3.2. Chương trình con:

- ♦ Định nghĩa chương trình con dạng hàm:

Function <Tên hàm>(<DS tham số>):<Kiểu hàm>;

Type <Cấu trúc cục bộ>

Var <Biến cục bộ>;

{

<Thân hàm>

}

III. Ngôn ngữ biểu diễn giải thuật

3.2. Chương trình con:

- ♦ Tham số chương trình con:
 - ✓ Viết cách nhau dấu “,”
 - ✓ Nếu có thêm từ khoá “var” đằng trước là tham biến, nếu không là tham trị.
- ♦ Gọi chương trình con:
 - ✓ Chương trình con được gọi bằng tên kèm theo danh sách tham số thực phù hợp.
 - ✓ Hàm được gọi trong biểu thức như một toán hạng
 - ✓ Thủ tục được gọi như câu lệnh độc lập

III. Ngôn ngữ biểu diễn giải thuật

3.3. Các lệnh:

- ♦ Lệnh gán:

$\langle \text{Tên biến} \rangle = \langle \text{Biểu thức} \rangle;$

Ví dụ: $a = a * 9$

- ♦ Lệnh ghép (khối lệnh):

$\{ \langle \text{Các lệnh} \rangle \}$

Ví dụ: $\{ d = d + 1; A = A * d \};$

Chú ý: các lệnh trong $\langle \text{Các lệnh} \rangle$ được viết cách nhau một dấu ;

III. Ngôn ngữ biểu diễn giải thuật

3.3. Các lệnh:

- ◆ Lệnh rẽ nhánh:

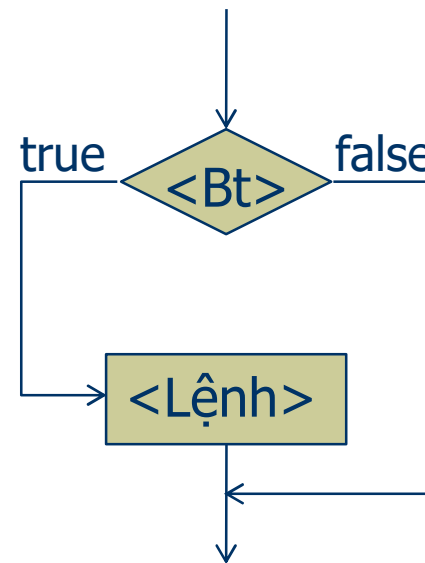
Kiểu thứ nhất:

If <Bt>
<Lệnh>;

Trong đó:

<Bt> là một biểu thức kiểu lô-gíc

<Lệnh> là một lệnh hay khối lệnh nào đó



III. Ngôn ngữ biểu diễn giải thuật

3.3. Các lệnh:

- ♦ Lệnh rẽ nhánh:

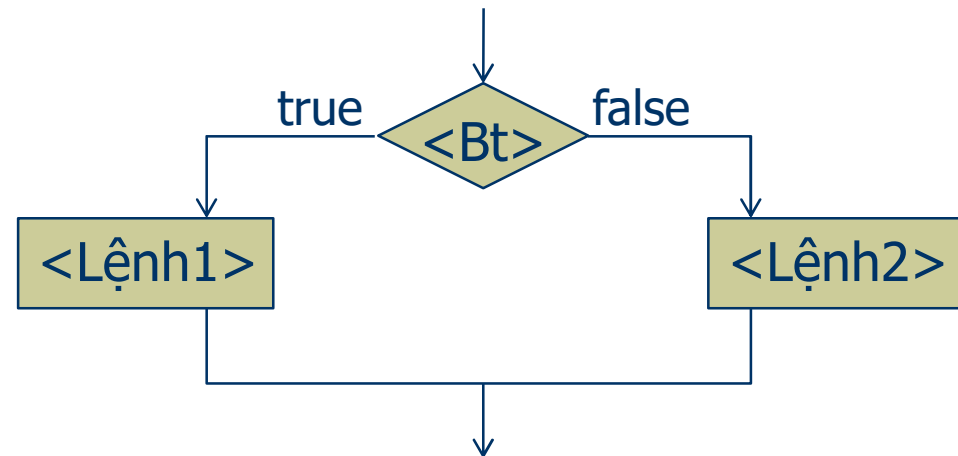
Kiểu thứ hai:

If <Bt>

<Lệnh1>

Else

<Lệnh2>;



III. Ngôn ngữ biểu diễn giải thuật

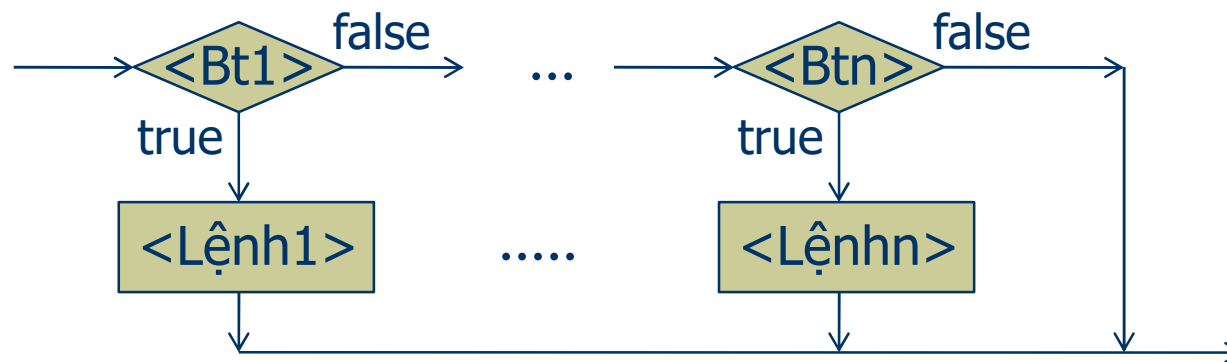
- ◆ Lệnh tuyển chọn: **If**

<Bt1> : <Lệnh1>

<Bt2> : <Lệnh2>

...

<Btn> : <Lệnhn> ;



III. Ngôn ngữ biểu diễn giải thuật

- ◆ Lệnh tuyển chọn:

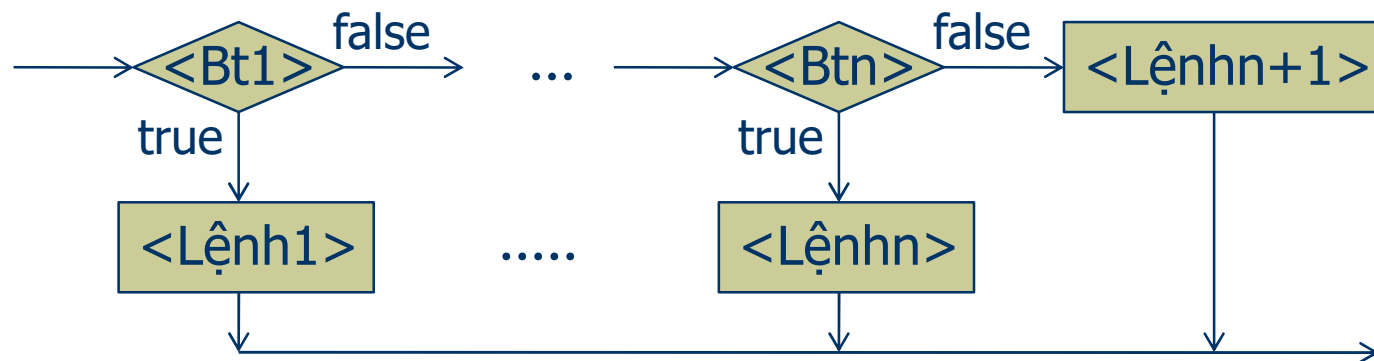
If

<Bt1> : <Lệnh1>;

...

<Btn> : <Lệnhn>

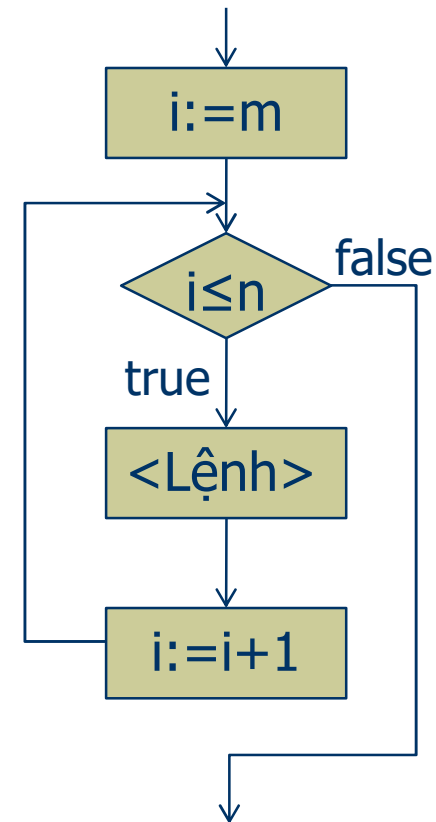
Else <Lệnhn+1>;



III. Ngôn ngữ biểu diễn giải thuật

- ♦ Vòng lặp xác định tiến:

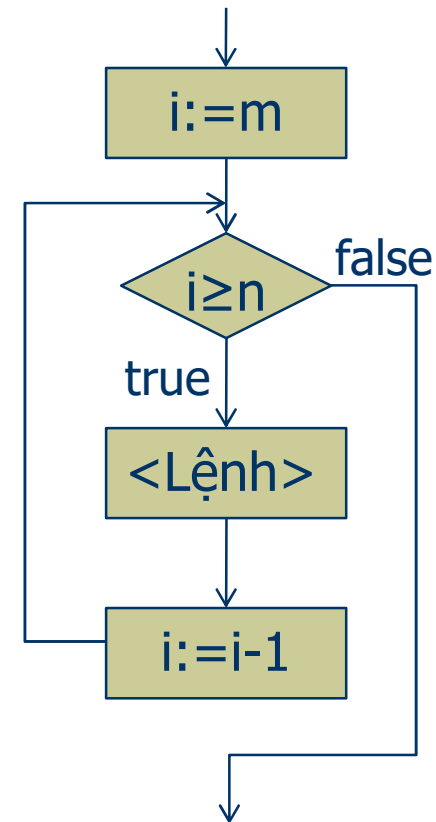
For ($i=m$; $i \leq n$; $i++$)
<Lệnh>;



III. Ngôn ngữ biểu diễn giải thuật

- ♦ Vòng lặp xác định lùi:

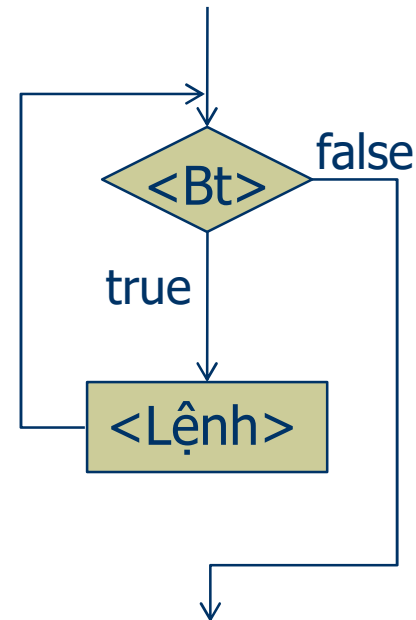
For ($i=m$; $i \leq n$; $i--$)
<Lệnh>;



III. Ngôn ngữ biểu diễn giải thuật

- ♦ Vòng lặp không xác định, điều kiện trước :

```
while <Bt>  
    <Lệnh>;
```



III. Ngôn ngữ biểu diễn giải thuật

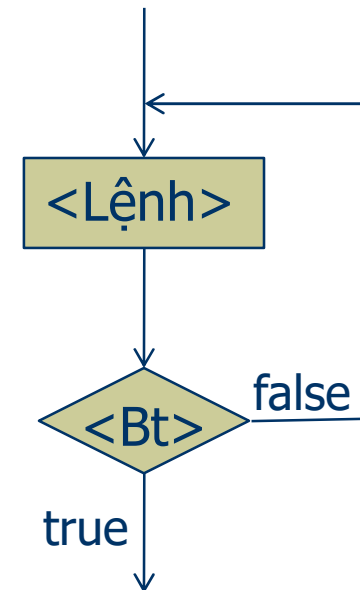
- ◆ Lệnh lặp:

Vòng lặp không xác định,
điều kiện sau:

do

<Lệnh>

while <Bt>;



III. Ngôn ngữ biểu diễn giải thuật

- ◆ Lệnh vào/ra:

Lệnh ghi ra màn hình:

Write (<Danh sách các biểu thức>);

Lệnh nhập dữ liệu từ bàn phím:

Read (<Danh sách biến>);

Trong đó: Danh sách các biểu thức & các biến viết cách nhau một dấu phẩy

III. Ngôn ngữ biểu diễn giải thuật

- ◆ Cấp phát động bộ nhớ:

Khai báo: \wedge <Kiểu dữ liệu>;

Xin cấp phát: New (<Tên biến>;

Giải phóng ô nhớ: Dispose (<Tên biến>;

Truy cập: <Tên biến> \wedge ;

Ví dụ: $p:\wedge\text{Int}$; {p là biến con trỏ trỏ vào ô nhớ kiểu Int}

New(p); {Xin cấp một biến nhớ kiểu Int có địa chỉ đặt trong p}

$p\wedge:=5$; {Gán giá trị cho ô nhớ được trỏ bởi p}

Dispose(p); {Giải phóng ô nhớ được trỏ bởi p}

<HẾT CHƯƠNG 1>