

 HOC VIEN NONG NGHIỆP VIỆT NAM
KHOA CÔNG NGHỆ THÔNG TIN


Chương 6 THUẬT TOÁN VÀ NGÔN NGỮ LẬP TRÌNH


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

NỘI DUNG

- 6.1. Phương pháp giải quyết vấn đề bằng máy tính
- 6.2. Thuật toán
- 6.3. Ngôn ngữ lập trình

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
2


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương


6.1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH

Nhắc lại:

- Một trong những chức năng cơ bản của máy tính: Xử lý thông tin đã nhận theo dãy lệnh đã nhớ sẵn bên trong
- Nguyên lý điều khiển bằng chương trình của Von Neumann: Máy tính hoạt động theo chương trình được lưu trữ sẵn trong bộ nhớ

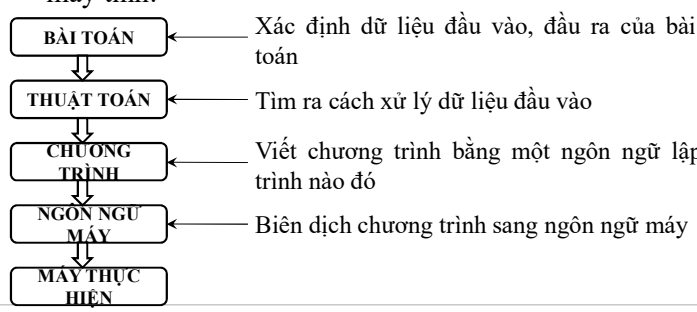
→ Để có thể giải quyết mỗi vấn đề/bài toán bằng máy tính thì cần phải xây dựng một chương trình máy tính tương ứng

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
3


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.1. PHƯƠNG PHÁP GIẢI QUYẾT VẤN ĐỀ BẰNG MÁY TÍNH


- Phương pháp chung để giải quyết vấn đề/bài toán bằng máy tính:



```

graph TD
    A[BÀI TOÁN] --> B[THUẬT TOÁN]
    B --> C[CHƯƠNG TRÌNH]
    C --> D[NGÔN NGỮ MÁY]
    D --> E[MÁY THỰC HIỆN]
    
```

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
4


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2. THUẬT TOÁN

6.2.1. Khái niệm thuật toán


6.2.2. Các tính chất của thuật toán

6.2.3. Cách diễn đạt thuật toán

6.2.4. Thiết kế thuật toán

6.2.5. Đánh giá thuật toán


08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 5


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.1. KHÁI NIỆM THUẬT TOÁN

- Thuật ngữ algorithm được đưa ra vào khoảng năm 825, xuất phát từ chữ algoritmi – phiên âm La tinh tên của nhà toán học người Trung Á Al-Khwarizmi
- Thuật toán (thuật giải, algorithms): là một dãy hữu hạn các thao tác, các phép toán có thể thực hiện được theo một trình tự xác định trên một số đối tượng dữ liệu nào đó để đạt được kết quả mong muốn
- Thuật toán được xây dựng phải bao gồm các thao tác được xác định rõ ràng, đơn giản và thực hiện được (phải “giao cho máy làm được”)
- Khi xây dựng một thuật toán cần xác định rõ thuật toán đó tác động lên dữ liệu nào


08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 6


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.1. KHÁI NIỆM THUẬT TOÁN

- Ví dụ: Bài toán tìm ước số chung lớn nhất của 2 số nguyên dương a và b:
 Input: a,b (nguyên dương)
 Output: (a,b)
- Thuật toán Euclid:
 - Bước 1: So sánh a và b, nếu $a = b$ thì dừng thuật toán và thông báo $(a,b) = b$. Nếu $a \neq b$ thì chuyển sang bước 2
 - Bước 2: Nếu $a > b$ thì thay thế a bởi a-b, nếu $a < b$ thì thay thế b bởi b-a. Quay lại thực hiện bước 1

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 7


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.2. CÁC TÍNH CHẤT CỦA THUẬT TOÁN

- Đầu vào
- Đầu ra
- Tính hữu hạn: Thuật toán phải kết thúc sau một số hữu hạn bước thực hiện
- Tính xác định
 - Mỗi bước của thuật toán phải được xác định chính xác, các thao tác được quy định chặt chẽ rõ ràng
 - Với cùng một dữ liệu đầu vào chỉ trả về 1 kết quả duy nhất
- Tính hiệu quả: Thuật toán đơn giản, dễ cài đặt, không gây tốn bộ nhớ, thực hiện nhanh

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 8

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.3. CÁCH DIỄN ĐẠT THUẬT TOÁN

3 cách:

- Cách 1: Liệt kê từng bước bằng ngôn ngữ tự nhiên:
- Sử dụng ngôn ngữ tự nhiên để liệt kê từng bước thực hiện của thuật toán với các quy tắc, thao tác cụ thể
- Ví dụ: Thuật toán Euclid tìm UCLN của 2 số nguyên dương a,b (slide 7)

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 9

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.3. CÁCH DIỄN ĐẠT THUẬT TOÁN

- Cách 2: Dùng lưu đồ:
- Sử dụng các hình khối cơ bản (Bắt đầu, Kết thúc, Khối Input, Khối Output, Khối điều kiện, Khối thao tác) và các cung để thể hiện các thao tác và trình tự thực hiện các thao tác của thuật toán

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 10

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

Các hình khối cơ bản để xây dựng lưu đồ

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 11

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

Lưu đồ thuật toán Euclid tìm (a,b)

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 12

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.3. CÁCH DIỄN ĐẠT THUẬT TOÁN

- Cách 3: Sử dụng giả mã (giả ngôn ngữ lập trình):
- Sử dụng các cấu trúc điều khiển của một ngôn ngữ lập trình kết hợp linh hoạt với ngôn ngữ tự nhiên và các ký hiệu toán học đơn giản nhằm diễn tả thuật toán
- Ví dụ: Giả mã cho thuật toán Euclid tìm (a,b) được viết tựa theo cấu trúc của ngôn ngữ lập trình PASCAL:


```
Nhập a,b
While a≠b do
    If a>b then thay a bởi a-b
    else thay b bởi b-a
Thông báo ước chung lớn nhất là b
```

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 13

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.3. CÁCH DIỄN ĐẠT THUẬT TOÁN

- Ví dụ (tiếp): Đoạn mã tương ứng viết bằng ngôn ngữ Pascal:


```
Writeln('Nhập 2 số nguyên dương a, b: ');
Write('a = '); Readln(a);
Write('b = '); Readln(b);
While a<>b do
    If a>b then a:=a-b
    else b:=b-a;
Writeln('Ước chung lớn nhất là ',b);
```

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 14

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.4. THIẾT KẾ THUẬT TOÁN

- Mô-đun hóa bài toán: Chia nhỏ bài toán (mô-đun chính) thành các bài toán nhỏ hơn (các mô-đun con)

```

graph TD
    A[A] --> a1[a1]
    A --> a2[a2]
    A --> a3[a3]
    a1 --> a11[a11]
    a1 --> a12[a12]
    a1 --> a13[a13]
    a3 --> a31[a31]
    a3 --> a32[a32]
  
```


08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 15

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.4. THIẾT KẾ THUẬT TOÁN

- Tinh chỉnh từng bước thuật toán:
 - Bước đầu thuật toán được minh họa bằng ngôn ngữ tự nhiên thể hiện các công việc chính cần thực hiện, sau đó dần minh họa chi tiết hơn với các thao tác xử lý, các phép toán được chỉ ra một cách cụ thể, đồng thời ngôn ngữ tự nhiên dùng để minh họa được thay thế dần bởi giả ngôn ngữ và ngày càng tiến gần đến ngôn ngữ lập trình
 - Trong quá trình thiết kế thuật toán, ngôn ngữ thể hiện dần được chuyển đổi theo sơ đồ: Ngôn ngữ tự nhiên → Giả ngôn ngữ → Ngôn ngữ lập trình


08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 16


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

Ví dụ

- Cho một dãy gồm n phần tử thuộc kiểu có thứ tự: a_1, a_2, \dots, a_n . Hãy đổi chỗ các phần tử trong dãy sao cho dãy sau khi đổi chỗ có thứ tự tăng dần

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
17


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương


Ví dụ

Ý tưởng:

- Chọn phần tử nhỏ nhất trong dãy nguồn rồi xếp vào vị trí đầu tiên trong dãy đích
- Chọn phần tử nhỏ nhất trong dãy nguồn còn lại (tức phần tử nhỏ thứ hai trong dãy nguồn ban đầu) rồi xếp vào vị trí thứ hai trong dãy đích
- ...

Lặp lại quá trình này cho đến khi hết dãy nguồn
 (Tổng quát: Tại bước thứ i , ta chọn ra phần tử nhỏ nhất trong dãy nguồn còn lại - tức phần tử nhỏ thứ i trong dãy nguồn ban đầu - rồi xếp vào vị trí thứ i trong dãy đích)

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
18


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương


Ví dụ

Giải mã dựa theo ngôn ngữ Pascal:

```

For i:=1 to n do
  Begin
    - Chọn phần tử nhỏ nhất  $a_j$  trong số các phần tử  $a_p, \dots, a_n$ 
    - Đổi chỗ  $a_j$  và  $a_i$  cho nhau
  End;
```

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
19


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương


Ví dụ

Các công việc trong khối Begin ... End sẽ được làm rõ hơn như sau:

```

j:=i;
For k:=i+1 to n do
  If  $a_k < a_j$  then j:=k;
Việc “đổi chỗ  $a_j$  và  $a_i$  cho nhau” được thực hiện bằng cách sử dụng thêm một phần tử trung gian min:
min:= $a_j$ ;
 $a_j$ := $a_i$ ;
 $a_i$ :=min;
```

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
20


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương


Ví dụ

Đoạn mã tương ứng viết bằng ngôn ngữ Pascal:

```

For i:=1 to n-1 do
  Begin
    j:=i;
    For k:=i+1 to n do
      If a[k]<a[j] then j:=k;
    If j<>i then
      Begin
        min:=a[j];
        a[j]:=a[i];
        a[i]:=min;
      End;
  End;
  
```

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
21


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

Ví dụ


Cho dãy số ban đầu:

3 6 -2 7 5

Dãy mới được sắp sau từng bước thực hiện thuật toán sắp xếp lựa chọn (i = 1..4):

i=1:	-2	6	3	7	5
i=2:	-2	3	6	7	5
i=3:	-2	3	5	7	6
i=4:	-2	3	5	6	7


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
22


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.5. ĐÁNH GIÁ THUẬT TOÁN

- Để đánh giá thuật toán có thể dựa trên nhiều tiêu chí: thời gian thực hiện thuật toán, khả năng thích ứng của thuật toán với các loại máy tính khác nhau, tính đúng đắn, mức độ đơn giản, hình thức của thuật toán, dung lượng bộ nhớ sử dụng để lưu trữ dữ liệu, ...
- 2 tiêu chí chính:
 - Thời gian thực hiện thuật toán
 - Dung lượng bộ nhớ sử dụng


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
23


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.5. ĐÁNH GIÁ THUẬT TOÁN

- Khi cài đặt thành chương trình máy tính, thời gian thực hiện của một thuật toán phụ thuộc vào rất nhiều yếu tố:
 - Số lượng các phép toán sơ cấp: các phép tính số học, các phép tính logic, các phép gán giá trị, chuyển chỗ, ... → phụ thuộc vào kích thước dữ liệu đầu vào của bài toán
 - Ngôn ngữ lập trình, chương trình dịch, hệ điều hành, tốc độ xử lý của máy tính, ... → những yếu tố này không đồng đều với mỗi loại máy tính, vì vậy không thể dùng chúng làm căn cứ để đánh giá thời gian thực hiện của thuật toán


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
24


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.5. ĐÁNH GIÁ THUẬT TOÁN

- Để đánh giá thời gian thực hiện của một thuật toán, người ta sử dụng “Độ phức tạp tính toán của thuật toán” (gọi tắt là Độ phức tạp của thuật toán): Thời gian thực hiện của thuật toán được đánh giá chỉ phụ thuộc vào kích thước của dữ liệu đầu vào, không phụ thuộc vào máy tính và các yếu tố liên quan

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
25



 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.5. ĐÁNH GIÁ THUẬT TOÁN

- Thuật toán T sử dụng để giải một bài toán có kích thước dữ liệu đầu vào n sẽ cần thực hiện T(n) các phép toán sơ cấp. T(n) là một hàm của tham số n, là đặc trưng cho độ phức tạp tính toán của thuật toán
- Tổng quát: Giả sử f(n), g(n) là hai hàm số không âm, đồng biến theo n. Hàm f(n) được xác định có độ phức tạp tính toán cấp g(n), ký hiệu là O(g(n)), khi và chỉ khi tồn tại các hằng số c và n₀ sao cho f(n) ≤ cg(n) khi n ≥ n₀. Khi đó, ta nói f(n) có cấp g(n), ký hiệu f(n) = O(g(n)) (thực chất là cấp lớn không vượt quá g(n))

Vi dụ: với f(n) = n² + 2n + 3, vì f(n) ≤ n² + 2n² + 3n² = 6n² với $\forall n \geq 1$ nên f(n) = O(n²)


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
26


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.5. ĐÁNH GIÁ THUẬT TOÁN

- Độ phức tạp tính toán của thuật toán có thể thuộc các dạng dưới đây (được sắp xếp theo mức độ tăng dần):
 - T(n) = O(1): độ phức tạp cấp hằng số
 - T(n) = O(log₂n): độ phức tạp cấp hàm logarit
 - T(n) = O(n): độ phức tạp cấp hàm tuyến tính
 - T(n) = O(nlog₂n): độ phức tạp cấp hàm nlog₂n
 - T(n) = O(n²), O(n³), ..., O(n^k): độ phức tạp cấp hàm đa thức
 - T(n) = O(2ⁿ), O(n!), O(nⁿ): độ phức tạp cấp hàm mũ
- Một thuật toán có độ phức tạp tính toán từ cấp hàm đa thức trở xuống thường chấp nhận được

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
27


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.5. ĐÁNH GIÁ THUẬT TOÁN

- Xác định độ phức tạp của thuật toán:


Quy tắc công:

Nếu T₁(n) = O(f(n)), T₂(n) = O(g(n)), thì T₁(n) + T₂(n) = O(max {f(n),g(n)})

Vi dụ: Trong một thuật toán có 3 bước, mỗi bước có độ phức tạp tính toán lần lượt là T₁(n) = O(n³), T₂(n) = O(n), T₃(n) = O(nlog₂n) thì thời gian thực hiện 3 bước là:

$$T_1(n) + T_2(n) + T_3(n) = O(\max \{n^3, n, n \log_2 n\}) = O(n^3)$$

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
28


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.5. ĐÁNH GIÁ THUẬT TOÁN

- Xác định độ phức tạp của thuật toán:


Quy tắc nhân:

Nếu $T_1(n) = O(f(n))$, $T_2(n) = O(g(n))$ thì: $T_1(n) \cdot T_2(n) = O(f(n) \cdot g(n))$

Ví dụ:

- Câu lệnh *For* $j:=1$ to n do $x:=x+1$; có thời gian thực hiện $T(n) = O(n \cdot 1) = O(n)$
- Câu lệnh *For* $i:=1$ to n do
 $\quad \quad \quad \text{For } j:=1$ to n do $x:=x+1$;
 có thời gian thực hiện $T(n) = O(n \cdot n) = O(n^2)$

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 29


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.2.5. ĐÁNH GIÁ THUẬT TOÁN

- Xác định độ phức tạp của thuật toán:


Quy tắc bỏ hằng số:

$O(c \cdot f(n)) = O(f(n))$ trong đó c là một hằng số

Ví dụ: $O(n^2/2) = O(n^2)$

- Lưu ý: Khi đánh giá độ phức tạp tính toán của thuật toán ta có thể chỉ cần quan tâm đến số lần thực hiện phép toán tích cực (active operation - phép toán mà số lần thực hiện nó không ít hơn số lần thực hiện của bất kỳ phép toán nào khác trong thuật toán)

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 30


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3. NGÔN NGỮ LẬP TRÌNH


6.3.1. Khái niệm về ngôn ngữ lập trình

6.3.2. Lịch sử phát triển của ngôn ngữ lập trình

6.3.3. Trình biên dịch và trình thông dịch

6.3.4. Các công việc của người lập trình


08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 31


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.1. KHÁI NIỆM VỀ NGÔN NGỮ LẬP TRÌNH

- Ngôn ngữ lập trình (programming language):
- Là ngôn ngữ dùng để viết các chương trình máy tính
- Bao gồm một hệ thống các ký hiệu, các từ khóa, các từ dành riêng (hay từ vựng), và các quy tắc để viết chương trình (hay cú pháp)


08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 32


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.2. LỊCH SỬ PHÁT TRIỂN CỦA NGÔN NGỮ LẬP TRÌNH

- Phân loại ngôn ngữ lập trình:
 - Ngôn ngữ máy
 - Hợp ngữ
 - Ngôn ngữ lập trình bậc cao


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
33


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.2. LỊCH SỬ PHÁT TRIỂN CỦA NGÔN NGỮ LẬP TRÌNH

- Ngôn ngữ máy:
 - Là ngôn ngữ duy nhất mà Bộ vi xử lý có thể nhận biết và thực hiện trực tiếp, các chương trình máy tính được viết bằng các ngôn ngữ khác phải được dịch sang ngôn ngữ máy trước khi thực thi
 - Lệnh máy được viết ở dạng số nhị phân hoặc biến thể của chúng trong hệ 16
 - Các chương trình thực hiện nhanh chóng, nhưng các lệnh máy dài và khó nhớ, chương trình cồng kềnh, mất thời gian khi viết và gây khó khăn cho việc đọc, phát hiện lỗi và hiệu chỉnh chương trình


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
34


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.2. LỊCH SỬ PHÁT TRIỂN CỦA NGÔN NGỮ LẬP TRÌNH

- Hợp ngữ:
 - Ra đời từ năm 1950 là ngôn ngữ lập trình bậc thấp
 - Cấu trúc lệnh rất giống với ngôn ngữ máy nhưng cho phép viết lệnh dưới dạng mã chữ, thường là những từ tiếng Anh viết tắt có ý nghĩa rõ ràng, dễ nhớ
 - Cho phép định địa chỉ hình thức
 - Các chương trình hợp ngữ được chuyển sang mã máy thông qua trình hợp dịch (assembler)
 - Gần với tầng thiết kế máy tính, các chương trình được viết luôn có sự liên quan chặt chẽ đến kiến trúc máy tính


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
35



 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương


6.3.2. LỊCH SỬ PHÁT TRIỂN CỦA NGÔN NGỮ LẬP TRÌNH


- Hợp ngữ (tiếp):
 - Hiện chỉ dùng khi cần lập trình thao tác trực tiếp với phần cứng máy tính hoặc làm các công việc không thường xuyên (trình điều khiển (driver), các hệ nhúng bậc thấp, các hệ thống thời gian thực, ...)


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
36

 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam Bài giảng Tin học đại cương		
6.3.2. LỊCH SỬ PHÁT TRIỂN CỦA NGÔN NGỮ LẬP TRÌNH		
<ul style="list-style-type: none"> • Ngôn ngữ lập trình bậc cao: - Là ngôn ngữ rất gần với ngôn ngữ tự nhiên và ngôn ngữ toán học - Thường sử dụng hệ thống ký hiệu phong phú với các ký hiệu số, các ký hiệu chữ, các ký hiệu toán học và nhiều ký hiệu thông dụng khác, cùng với các từ khóa tiếng Anh đơn giản, các cấu trúc lệnh chặt chẽ, rõ ràng và mang ý nghĩa thực tế - Dễ học, dễ đọc, dễ viết và hiệu chỉnh chương trình → cho phép thể hiện chính xác các thuật toán, có tính độc lập cao, ít phụ thuộc vào phần cứng máy tính 		
08/02/2017	Chương 6. Thuật toán và Ngôn ngữ lập trình	37

 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam Bài giảng Tin học đại cương		
6.3.2. LỊCH SỬ PHÁT TRIỂN CỦA NGÔN NGỮ LẬP TRÌNH		
<ul style="list-style-type: none"> • Ngôn ngữ lập trình bậc cao (tiếp): - Còn được gọi là ngôn ngữ thuật toán - Các chương trình muốn máy tính thực thi được thì cần phải được dịch sang ngôn ngữ máy nhờ các chương trình dịch - Ví dụ: Fortran, Pascal, C, C++, Java, PHP, ... 		
08/02/2017	Chương 6. Thuật toán và Ngôn ngữ lập trình	38

 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam Bài giảng Tin học đại cương		
6.3.2. LỊCH SỬ PHÁT TRIỂN CỦA NGÔN NGỮ LẬP TRÌNH		
<ul style="list-style-type: none"> • Hiện nay, việc phân loại ngôn ngữ lập trình chỉ mang tính tương đối. Tùy theo từng mục đích, có thể phân loại ngôn ngữ lập trình theo những cách khác nhau. Ví dụ: - Phân loại theo mức trừu tượng: có nhóm ngôn ngữ lập trình bậc thấp và nhóm ngôn ngữ lập trình bậc cao - Phân loại theo hình thức lập trình: có nhóm ngôn ngữ khai báo (LIST, PROLOG, ...) và nhóm ngôn ngữ mệnh lệnh (PASCAL, C, ...) - Phân loại theo các họ, có họ ngôn ngữ máy và hợp ngữ, họ ngôn ngữ cổ điển (ALGOL, PASCAL, C, ...), họ ngôn ngữ hàm (LISP, ...), họ ngôn ngữ logic (PROLOG, ...), họ ngôn ngữ hướng đối tượng (C++, JAVA, ...), họ ngôn ngữ truy vấn (SQL, ...) 		
08/02/2017	Chương 6. Thuật toán và Ngôn ngữ lập trình	39


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam Bài giảng Tin học đại cương		
6.3.2. LỊCH SỬ PHÁT TRIỂN CỦA NGÔN NGỮ LẬP TRÌNH		
<ul style="list-style-type: none"> • Một số ngôn ngữ lập trình tiêu biểu: - FORTRAN, ALGOL, LISP, COBOL, BASIC, PASCAL, C, C++, PERL, PYTHON, RUBBY, JAVA, PHP, ... 		
08/02/2017	Chương 6. Thuật toán và Ngôn ngữ lập trình	40


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.3. TRÌNH BIÊN DỊCH VÀ TRÌNH THÔNG DỊCH

- Máy tính chỉ hiểu được một ngôn ngữ duy nhất là ngôn ngữ máy. Trước khi được thực thi, các chương trình viết bằng các ngôn ngữ lập trình không phải là ngôn ngữ máy (chương trình nguồn) phải được dịch sang ngôn ngữ máy nhờ các chương trình dịch
- 2 loại chương trình dịch:
 - Trình thông dịch
 - Trình biên dịch


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
41


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.3. TRÌNH BIÊN DỊCH VÀ TRÌNH THÔNG DỊCH

- Trình thông dịch: Sử dụng kỹ thuật thông dịch, dịch từng câu lệnh trong chương trình nguồn được viết bằng ngôn ngữ lập trình bậc cao sang ngôn ngữ máy để máy tính “hiểu” và thực thi ngay câu lệnh đó mà không lưu lại đoạn mã máy tương ứng, sau đó chuyển sang dịch câu lệnh tiếp theo
- Không tạo ra tệp mã đối tượng (tệp mã máy tương ứng với chương trình nguồn). Mỗi lần thực hiện chương trình là một lần thông dịch lại


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
42


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.3. TRÌNH BIÊN DỊCH VÀ TRÌNH THÔNG DỊCH

- Trình thông dịch (tiếp):
- Cho phép dịch, thực hiện ngay câu lệnh mà không cần phải đợi dịch xong toàn bộ chương trình, cho phép dò tìm lỗi dễ dàng → thích hợp trong môi trường cần có sự đối thoại giữa con người và hệ thống
- Một số ngôn ngữ lập trình có sử dụng trình thông dịch như: BASIC, VISUAL BASIC, PERL, PYTHON, ...


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
43


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.3. TRÌNH BIÊN DỊCH VÀ TRÌNH THÔNG DỊCH

- Trình biên dịch (Compiler): Sử dụng kỹ thuật biên dịch, dịch toàn bộ chương trình nguồn sang ngôn ngữ máy và tạo ra tệp mã đối tượng tương ứng
- Trong quá trình biên dịch, trình biên dịch phân tích từ vựng và cú pháp của các câu lệnh, thông báo danh sách tất cả các lỗi để lập trình viên chỉnh sửa. Tệp mã đối tượng chỉ được tạo ra khi chương trình nguồn không còn bất kỳ lỗi cú pháp nào
- Mỗi lần thực hiện chương trình chỉ cần sử dụng chương trình thực thi đã được tạo trước đó mà không cần phải tiến hành biên dịch lại chương trình nguồn
- thích hợp với các chương trình có tính ổn định và được thực hiện nhiều lần


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
44


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.3. TRÌNH BIÊN DỊCH VÀ TRÌNH THÔNG DỊCH

- Trình biên dịch (tiếp):
- Thông thường, mỗi ngôn ngữ lập trình bậc cao đều có một trình biên dịch tương ứng, ví dụ: PASCAL, C, C++, ...


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
45


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.4. CÁC CÔNG VIỆC CỦA LẬP TRÌNH

- Bước 1: Soạn thảo chương trình:
- Sử dụng một trình soạn thảo chuyên dụng để nhập nội dung chương trình, lưu tệp chương trình (tệp mã nguồn - source code) với phần mở rộng tên tệp phù hợp với ngôn ngữ lập trình (ví dụ: .pas, .c, .cpp, ...)


08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
46


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.4. CÁC CÔNG VIỆC CỦA LẬP TRÌNH

- Bước 2: Biên dịch chương trình:
- Sử dụng trình biên dịch (compiler) thích hợp để biên dịch tệp chương trình nguồn sang tệp mã máy tương ứng (tệp đối tượng hay object code). Nếu chương trình nguồn có một số lỗi nào đó về mặt cú pháp thì trình biên dịch sẽ thông báo danh sách tất cả các lỗi, khi đó cần quay lại bước 1, sử dụng trình soạn thảo để chỉnh sửa chương trình nguồn
- Khi tệp đối tượng đã được tạo, bộ liên kết (linker) sẽ thực hiện việc liên kết các đối tượng thành phần với nhau và tạo ra tệp thực thi (executable code) cho chương trình

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
47


 Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.4. CÁC CÔNG VIỆC CỦA LẬP TRÌNH

- Bước 3: Chạy thử chương trình
- Chạy chương trình (kích hoạt tệp thực thi), nhập các dữ liệu đầu vào (các dữ liệu mẫu dùng để kiểm tra) và kiểm tra các kết quả được đưa ra. Nếu kết quả thu được không đúng hoặc có lỗi khi thực thi chương trình thì cần kiểm tra, chỉnh sửa lại thuật toán, rồi quay lại bước 1 để chỉnh sửa lại chương trình

08/02/2017
Chương 6. Thuật toán và Ngôn ngữ lập trình
48

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

6.3.4. CÁC CÔNG VIỆC CỦA LẬP TRÌNH

- Môi trường phát triển tích hợp (IDE – Integrated Development Environment): Tích hợp trình soạn thảo, trình biên dịch, bộ liên kết, trình gỡ rối, ... và cho phép chạy thử chương trình
- Người lập trình cũng có thể sử dụng một trình soạn thảo chuyên dụng, độc lập để soạn thảo chương trình nguồn (Notepad++, ...); sau đó sử dụng một trình biên dịch thích hợp để biên dịch rồi chạy chương trình bằng cách kích hoạt tệp thực thi đã được tạo

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 49

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

Ví dụ

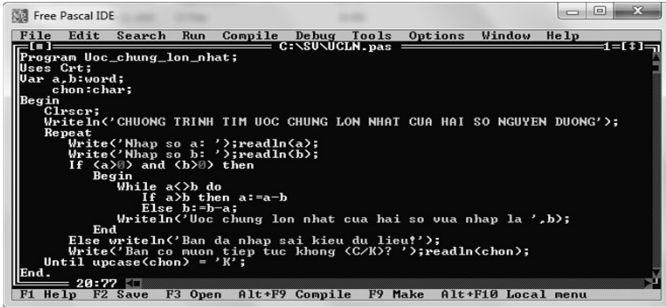
- Viết chương trình tìm ước số chung lớn nhất của 2 số nguyên dương, chương trình viết bằng ngôn ngữ PASCAL, sử dụng phần mềm Free Pascal (IDE, version 2.6.2) để lập trình

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 50

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

Ví dụ

- Bước 1: Khởi động phần mềm Free Pascal, sử dụng trình soạn thảo nhập nội dung chương trình nguồn, sau đó lưu tệp mã nguồn dưới dạng .pas:



```

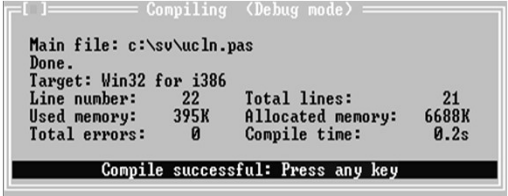
Program Uoc_chung_lon_nhat;
Uses Crt;
Var a,b:word;
    chon:char;
Begin
  Clrscr;
  WriteLn<'CHƯƠNG TRÌNH TÌM ƯOC CHUNG LON NHAT CỦA HAI SỐ NGUYÊN DƯƠNG'>;
  Repeat
    Write<'Nhập số a: '>;readln(a);
    Write<'Nhập số b: '>;readln(b);
    If (a>0) and (b>0) then
      Begin
        While a<>b do
          If a>b then a:=a-b
          Else b:=b-a;
        WriteLn<'Ưoc chung lon nhât của hai số vừa nhập là ',b>;
      End
    Else writeLn<'Bạn đã nhập sai kiểu dữ liệu!>;
    Write<'Bạn có muốn tiếp tục không <C/K? '>;readln(chon);
  Until upcase(chon) = 'K';
End.
  
```

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 51

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

Ví dụ

- Bước 2: Nhấn tổ hợp phím Alt+F9 để biên dịch chương trình. Khi chương trình nguồn không có lỗi cú pháp, hệ thống sẽ đưa ra thông báo quá trình biên dịch đã thành công:



```


[ ] Compiling (Debug mode)
Main file: c:\sv\uecln.pas
Done.
Target: Win32 for i386
Line number: 22 Total lines: 21
Used memory: 395K Allocated memory: 6688K
Total errors: 0 Compile time: 0.2s
Compile successful: Press any key
  
```

08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 52

Khoa Công nghệ thông tin – Học viện Nông nghiệp Việt Nam
Bài giảng Tin học đại cương

Ví dụ

- Bước 3: Nhấn tổ hợp phím Ctrl+F9 để chạy thử chương trình:



08/02/2017 Chương 6. Thuật toán và Ngôn ngữ lập trình 53