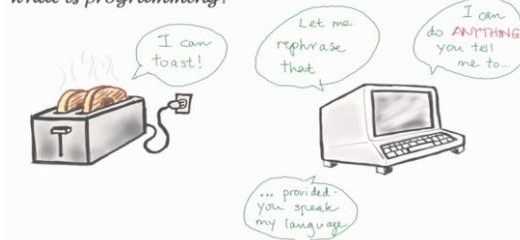


What is programming?



Tuần 15

THUẬT TOÁN và LƯU ĐỒ

CT101 – Lập trình căn bản

Khoa CNTT&TT – ĐHCT

Mục đích

Nhằm giới thiệu đến sinh viên khái niệm thuật toán (algorithms), các phương pháp trình bày thuật toán và bước đầu hình thành kỹ năng thiết kế thuật toán, một khâu then chốt trong quy trình xây dựng phần mềm

Một thuật toán có nhiều ứng dụng cũng được trình bày trong tuần này là thuật toán tìm giá trị lớn nhất, giá trị nhỏ nhất

Yêu cầu

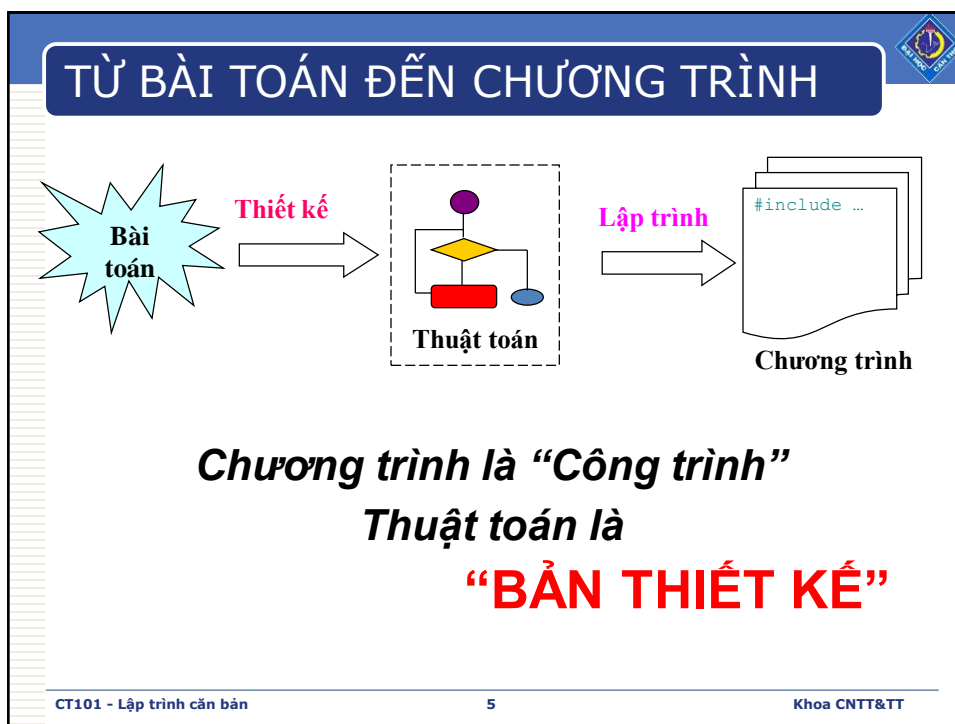


- Hiểu được khái niệm thuật toán.
- Hiểu được vai trò của thuật toán trong lập trình giải quyết vấn đề.
- Hiểu được các phương pháp biểu diễn thuật toán
- Vận dụng được lưu đồ để biểu diễn một số thuật toán cơ bản.
- Hiểu được thuật toán và viết được chương trình tìm giá trị lớn nhất, giá trị nhỏ nhất của một mảng

Nội dung



1. Từ bài toán đến chương trình
2. Khái niệm thuật toán
3. Các đặc trưng của thuật toán
4. Ngôn ngữ biểu diễn thuật toán
5. Một số cấu trúc suy luận cơ bản
6. Thuật toán và chương trình tìm giá trị lớn nhất, giá trị nhỏ nhất của một mảng



KHÁI NIỆM THUẬT TOÁN

- **Ví dụ:** Hoán đổi chất lỏng trong 2 bình A (đựng nước mắm) và B (đựng rượu):
 - Yêu cầu phải có thêm một bình thứ ba, bình C.
 - **Bước 1:** Đổ rượu từ bình A sang bình C.
 - **Bước 2:** Đổ nước mắm từ bình B sang bình A.
 - **Bước 3:** Đổ rượu từ bình C sang bình B.
- Thuật toán (Algorithms) là **một dãy các thao tác** theo trật tự nhất định để giải quyết một vấn đề.

CT101 - Lập trình căn bản 6 Khoa CNTT&TT

CÁC ĐẶC TRƯNG CỦA THUẬT TOÁN



- **Tính kết thúc/ Tính hữu hạn**
 - Một thuật toán bao giờ cũng phải kết thúc sau hữu hạn bước mặc dù số bước này có thể rất lớn.
- **Tính xác định**
 - Mọi bước của một thuật toán bao giờ cũng được xác định rõ ràng, chính xác và do đó luôn thực hiện được.
- **Có đầu vào, đầu ra**
 - Mỗi thuật toán đều có một số giá trị nhận vào (Input values, đơn giản là Input).
 - Mỗi thuật toán có một số giá trị đưa ra (Output values, đơn giản là Output).

CÁC ĐẶC TRƯNG CỦA THUẬT TOÁN



- **Tính tổng quát**
 - Thuật toán không chỉ áp dụng cho một bài toán nhất định mà có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.
- **Tính hiệu quả**
 - Thời gian
 - Tài nguyên máy
 - Độ phức tạp của thuật toán

NGÔN NGỮ BIỂU DIỄN THUẬT TOÁN



- Ngôn ngữ tự nhiên
- Mã giả (Pseudocode)
- Ngôn ngữ sơ đồ - Lưu đồ (Flowcharts)

NGÔN NGỮ TỰ NHIÊN



- Là ngôn ngữ sử dụng trong đời sống.
- Thuật toán sẽ được trình bày bằng cách mô tả các bước thực hiện.
- Ví dụ: Thuật toán giải phương trình bậc nhất $ax+b=0$.

Bước 1: Nhận giá trị của các tham số ***a, b***.

Bước 2: Xét giá trị của ***a*** xem có bằng 0 hay không?

Nếu $a=0$ thì làm bước 3, nếu a khác không thì làm bước 4.

Bước 3: (a bằng 0) Nếu ***b*** bằng 0 \Rightarrow pt vô số nghiệm.

Nếu ***b*** khác 0 \Rightarrow pt vô nghiệm.

Bước 4: (a khác 0) \Rightarrow phương trình có nghiệm $x=-b/a$.

MÃ GIẢ (PSEUDOCODE)








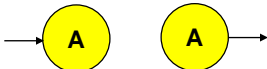
- Là một sự kết hợp giữa ngôn ngữ tự nhiên với các cấu trúc lệnh của một ngôn ngữ lập trình.
- Thuật toán trình bày bằng mã giả sẽ gần với chương trình nhưng vẫn giữ được sự “tự nhiên” trong suy nghĩ của người lập trình.
- Người lập trình sẽ tinh chế từng bước để có CT
- **Ví dụ:**
 - Trong khi i còn là chỉ số của mảng và chưa tìm thấy
Nếu $a[i]$ bằng X thì tìm thấy; ngược lại tăng i lên 1 đơn vị
 - while ($i < n$) && (!found)
if ($a[i] == X$) found = 1; else $i++$;

LƯU ĐỒ (FLOWCHART)



- Các bước của thuật toán được thể hiện bởi các hình khối, nối với nhau bởi các mũi tên. Do đó còn gọi là ngôn ngữ sơ đồ khối.
- Thuật toán được trình bày một cách ngắn gọn, làm nổi bật các cấu trúc suy luận, dễ theo dõi quá trình xử lý.
- Biểu diễn cho các thuật toán lớn, phức tạp.

HÌNH KHỐI DÙNG TRONG LƯU ĐỒ

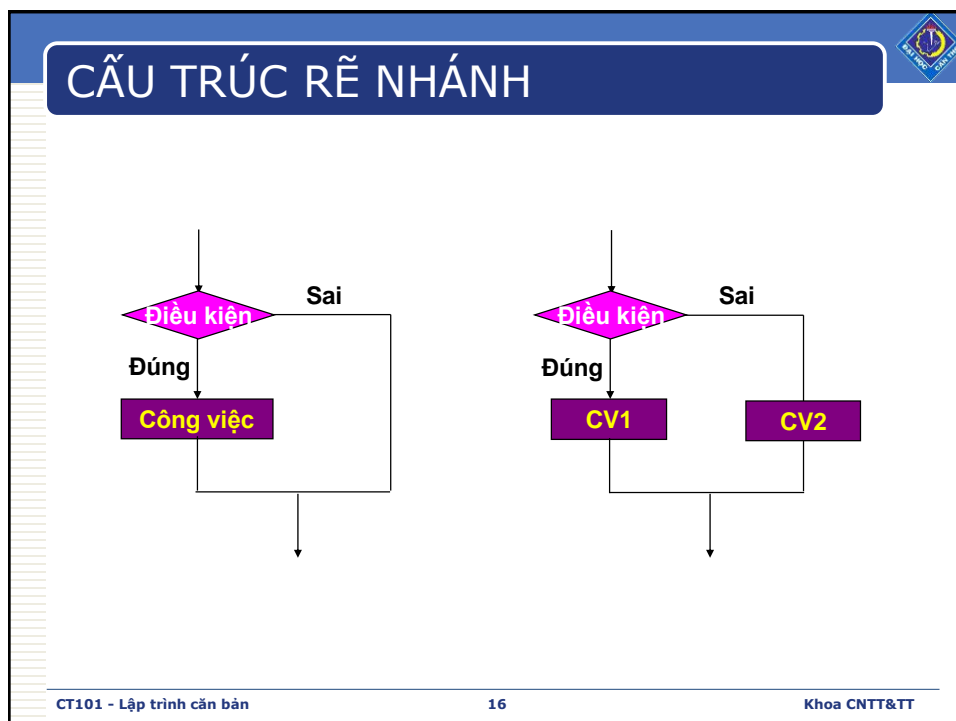
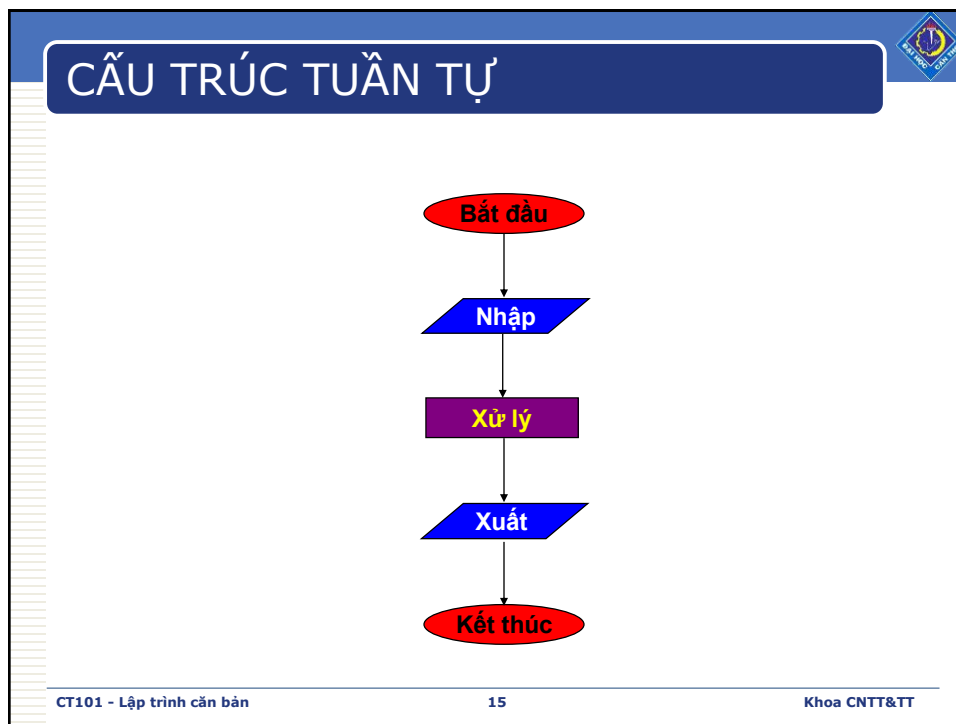
Hình khối	Ý nghĩa
	Bắt đầu hoặc kết thúc
	Nhập hoặc xuất
	Tính toán hoặc xử lý
	Quyết định hoặc chọn lựa
	Đường đi chỉ trình tự thực hiện
	Điểm nối để nối các phần khác nhau của một lưu đồ lại với nhau.

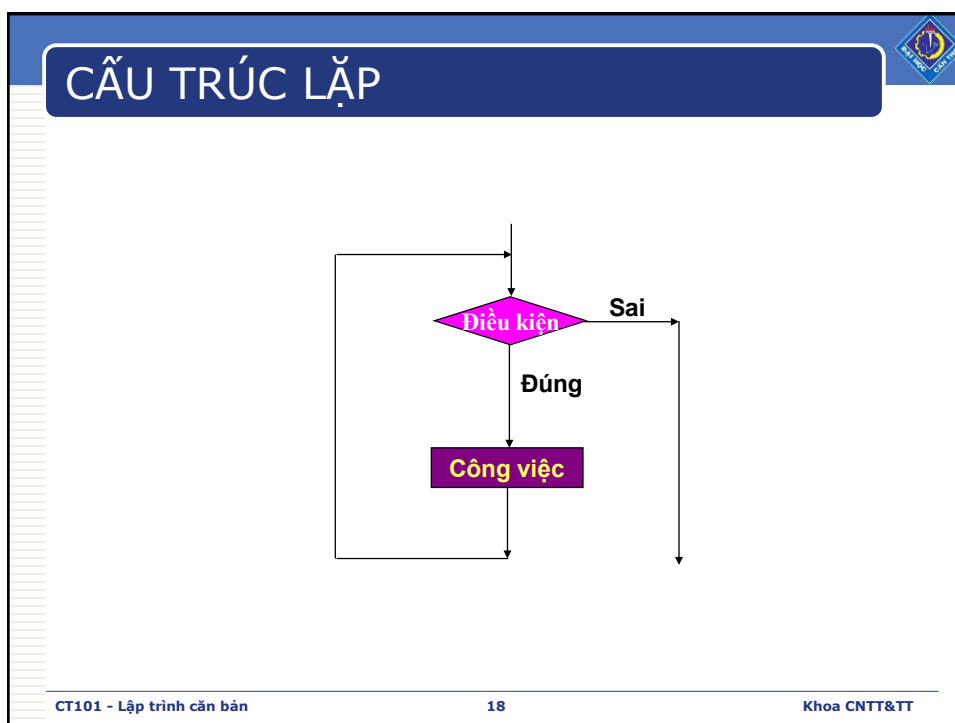
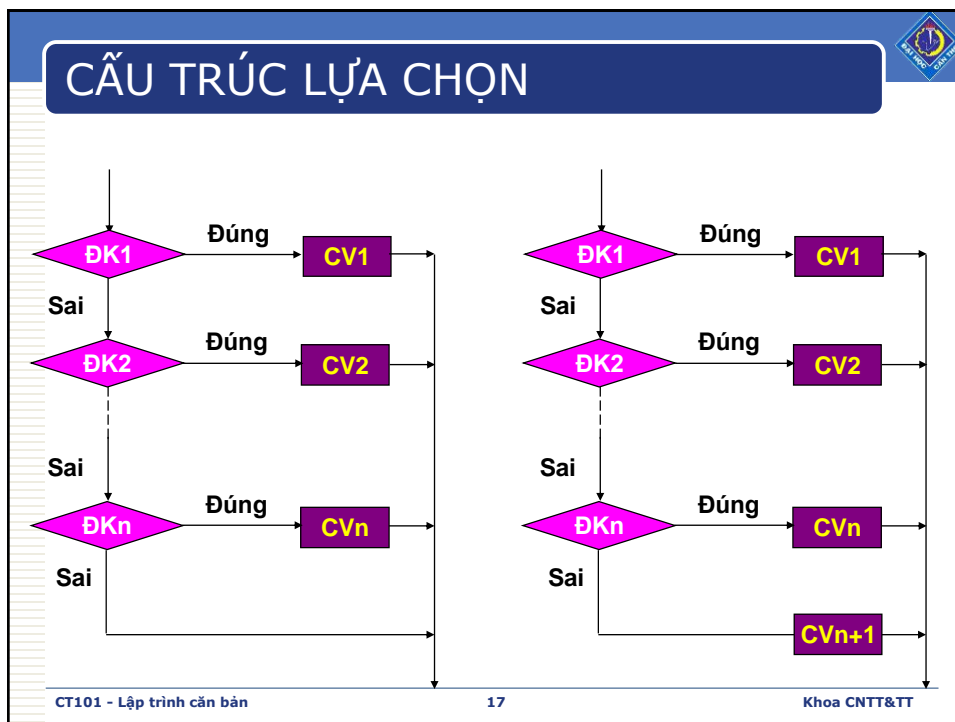
CT101 - Lập trình căn bản 13 Khoa CNTT&TT

MỘT SỐ CẤU TRÚC SUY LUẬN CƠ BẢN

- Tuần tự
- Rẽ nhánh
- Lựa chọn
- Vòng lặp

CT101 - Lập trình căn bản 14 Khoa CNTT&TT





GHI NHỚ



Thiết kế thuật toán

TRƯỚC KHI

Lập trình

TÌM GIÁ TRỊ LỚN NHẤT CỦA MỘT MẢNG



- Bài toán: Cho một mảng a , có n phần tử là các giá trị (thuộc một kiểu có thứ tự). Tìm giá trị lớn nhất của mảng
- Ý tưởng:
 - Khởi tạo $MaxValue = a[0]$
 - Xét các phần tử i , bắt đầu từ 1 đến $n-1$, nếu $a[i] > MaxValue$ thì đặt lại $MaxValue = a[i]$
 - Trả về $MaxValue$
- $DataType\ Max(DataType\ a[],\ int\ n)$
 - $DataType$ là một kiểu có thứ tự như các kiểu số, kiểu chuỗi ký tự

TÌM GIÁ TRỊ LỚN NHẤT CỦA MỘT MẢNG



```
#include <stdio.h>

typedef int DataType;
DataType Max(DataType a[], int n){
    DataType somax =a[0];
    int i;
    for(i=1; i<n; i++)
        if(a[i]>somax) somax=a[i];
    return somax;
}

int main(){
    DataType a[] = {1, 5, 7, 12, 4, 6};
    printf("Gia tri lon nhat la %d\n", Max(a,6));
    return 0;
}
```

Tìm GTLN: Khởi tạo somax = $-\infty$



```
#include <stdio.h>
#include <limits.h>

typedef int DataType;
DataType Max(DataType a[], int n){
    DataType somax =INT_MIN;// So nguyen nho nhat
    int i;
    for(i=1; i<n; i++)
        if(a[i]>somax) somax=a[i];
    return somax;
}

int main(){
    DataType a[] = {1, 5, 7, 12, 4, 6};
    printf("Gia tri lon nhat la %d\n", Max(a,6));
    return 0;
}
```

Định vị GTLN: Trả về vị trí đạt GTLN

```
#include <stdio.h>
typedef int DataType;

int LocateMax(DataType a[], int n){
    DataType somax= a[0];
    int i, index= 0;
    for(i=1; i<n; i++){
        if(a[i]>somax){
            somax= a[i];
            index= i;
        }
    }
    return index;
}

int main(){
    DataType a[] = {1, 5, 7, 12, 4, 6};
    printf("Phan tu dat GTLN la %d\n", LocateMax(a,6));
    return 0;
}
```

Tổng kết

- Không thể có chương trình, nếu không có thuật toán
- Một trong những đặc trưng quan trọng nhất của thuật toán là tính tổng quát
- Thuật toán có thể biểu diễn bằng ngôn ngữ tự nhiên, mã giả hoặc lưu đồ.
- Các cấu trúc suy luận cơ bản là: tuần tự, rẽ nhánh, lựa chọn và vòng lặp
- Bài toán tìm GTLN/GTNN có rất nhiều ứng dụng sau này nên phải nắm vững

Kiểm tra kiến thức







1. Thuật toán là

- một dãy các thao tác theo trật tự nhất định để giải quyết một vấn đề
- một dãy các thao tác theo cấu trúc tuần tự, rẽ nhánh, lựa chọn và vòng lặp
- một dãy các thao tác theo trật tự nhất định và có thể được biểu diễn bởi ngôn ngữ tự nhiên, mã giả hoặc lưu đồ.
- một dãy các thao tác có tính tổng quát để giải quyết một vấn đề

Kiểm tra kiến thức



2. Hãy bắt cặp các chữ cái và các số cho phù hợp

1		A	Tính toán hoặc xử lý
2		B	Quyết định hoặc chọn lựa
3		C	Bắt đầu hoặc kết thúc
4		D	Nhập hoặc xuất

Bài tập tổng kết: Thiết kế các thuật toán



1. Tính tổng của n số nhập từ bàn phím
2. Tính $n!$ theo định nghĩa và theo công thức đệ quy
3. Tính x^n
4. Tìm ước số chung lớn nhất của 2 số a, b
5. Xét xem n có phải là số nguyên tố
6. Xác định ma trận chuyển vị của một ma trận cấp m, n
7. Tính tổng của 2 ma trận cùng cấp
8. Tính tích của 2 ma trận cấp m, k và k, n

Bài tập tổng kết: Viết các hàm



9. Tìm giá trị nhỏ nhất của một mảng
10. Định vị giá trị nhỏ nhất của một mảng
11. Tìm GTLN/GTNN của một ma trận (mảng 2 chiều)
12. Cho mỗi sinh viên được biểu diễn bởi một cấu trúc, bao gồm các trường lưu trữ Mã SV, Họ tên SV và Điểm TB tích lũy. Danh sách SV là một mảng động của các SV. Hãy viết các hàm:
 1. Cho phép nhập vào một danh sách SV.
 2. Cho phép in một danh sách SV.
 3. Hàm nhận vào một danh sách có n SV và trả về SV có Điểm TB tích lũy lớn nhất
 4. Cho phép in SV có Điểm TB tích lũy lớn nhất



CT101 – Lập trình căn bản


Khoa CNTT&TT – ĐHCT