



# CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

<http://fit.vimaru.edu.vn>

LOGO

# Các vấn đề chung

**Bộ môn phụ trách:**

**BM Khoa học máy tính, Khoa Công nghệ thông tin.**

**Mã học phần:**

**17233**

**Số tín chỉ:**

**03**

**Hình thức thi học kỳ:**

**Thi viết, thời gian 75 phút. Điểm  $Z=0,5X+0,5Y$ .**

**Điều kiện tiên quyết:**

**SV phải hoàn thành các học phần: LTC, TRR**



# ***Cách tính điểm X***

- ❖ Điểm X1: đi học và làm bài tập đầy đủ hay không (=0 nếu đi học ít hơn 75% hoặc không làm bài tập đầy đủ theo yêu cầu);
- ❖ Điểm X2 là trung bình bài kt số 1 và số 2, X2 phải  $\geq 4.0$ ;
- ❖ Điểm X3: là bài kiểm tra số 3 (điểm kiểm tra thực hành), X3 phải  $\geq 4.0$ ;
- ❖  $X = 0.2 * X1 + 0.4 * X2 + 0.4 * X3$ .

# Mục tiêu của học phần

- ❖ Cung cấp các kiến thức cơ bản về thuật toán, cấu trúc dữ liệu.
- ❖ Cung cấp các kiến thức về chiến lược xây dựng và đánh giá thuật toán.
- ❖ Rèn luyện tư duy khoa học trong lập trình.



# Nội dung chính



**Chương 1.** Các khái niệm cơ bản

**Chương 2.** Các cấu trúc dữ liệu cơ bản

**Chương 3.** Tìm kiếm và sắp xếp

**Chương 4.** Cấu trúc cây





# Chương 1. Các khái niệm cơ bản

## ❖ Nội dung:

- ✓ Khái niệm thuật toán, cấu trúc dữ liệu.
- ✓ Cách biểu diễn thuật toán.
- ✓ Đánh giá độ phức tạp thuật toán.
- ✓ Mối quan hệ giữa cấu trúc dữ liệu và giải thuật
- ✓ ...



## *1.1. Giới thiệu về thuật toán*

- ❖ Khái niệm: Một thuật toán là một thủ tục tính toán xác định (well-defined) nhận một tập các giá trị gọi là đầu vào (input) và sinh ra một tập giá trị được gọi là đầu ra (output)”.
- ❖  $\Rightarrow$  Thuật toán giống như là các cách thức, qui trình để hoàn thành một công việc cụ thể xác định (well-defined) nào đó



## *1.2. Giới thiệu về thuật toán*

- ❖ Thuật toán là một dãy hữu hạn các bước, mỗi bước mô tả chính xác các phép toán hoặc hành động cần thực hiện, để giải quyết một (số) vấn đề.
- ❖ (Từ điển Oxford Dictionary định nghĩa, “Algorithm: set of well - defined rules for solving a problem in a finite number of steps.”)





# *Giới thiệu về thuật toán – ví dụ*

- ❖ Tìm ước số chung lớn nhất của 2 số nguyên dương  $a, b$ .
  - ✓ Input: Hai số nguyên dương  $a$  và  $b$ .
  - ✓ Output: USCLN của  $a$  và  $b$



# *Đặc trưng của thuật toán*

- 1 Tính đúng đắn
- 2 Tính dừng
- 3 Tính xác định
- 4 Tính hiệu quả
- 5 Tính phổ quát
- 6 Dữ liệu vào và dữ liệu ra



# *Biểu diễn thuật toán*

- 1 Mô tả các bước thực hiện thuật toán bằng ngôn ngữ tự nhiên hoặc lập trình
- 2 Mô tả thuật toán bằng sơ đồ khối



# *Mô tả các bước thực hiện*

Sử dụng ngôn ngữ tự nhiên, ngôn ngữ lập trình, hoặc giả mã để mô tả các bước thực hiện thuật toán

**Ví dụ:** mô tả thuật toán tìm ước số chung lớn nhất của hai số nguyên dương.

**Input:** Hai số nguyên dương  $a, b$ .

**Output:** Ước số chung lớn nhất của  $a, b$ .

1

Nếu  $a=b$  thì  $USCLN(a, b)=a$ .

2







Nếu  $a > b$  thì tìm  $USCLN$  của  $a-b$  và  $b$ , quay lại bước 1;

3

Nếu  $a < b$  thì tìm  $USCLN$  của  $a$  và  $b-a$ , quay lại bước 1;

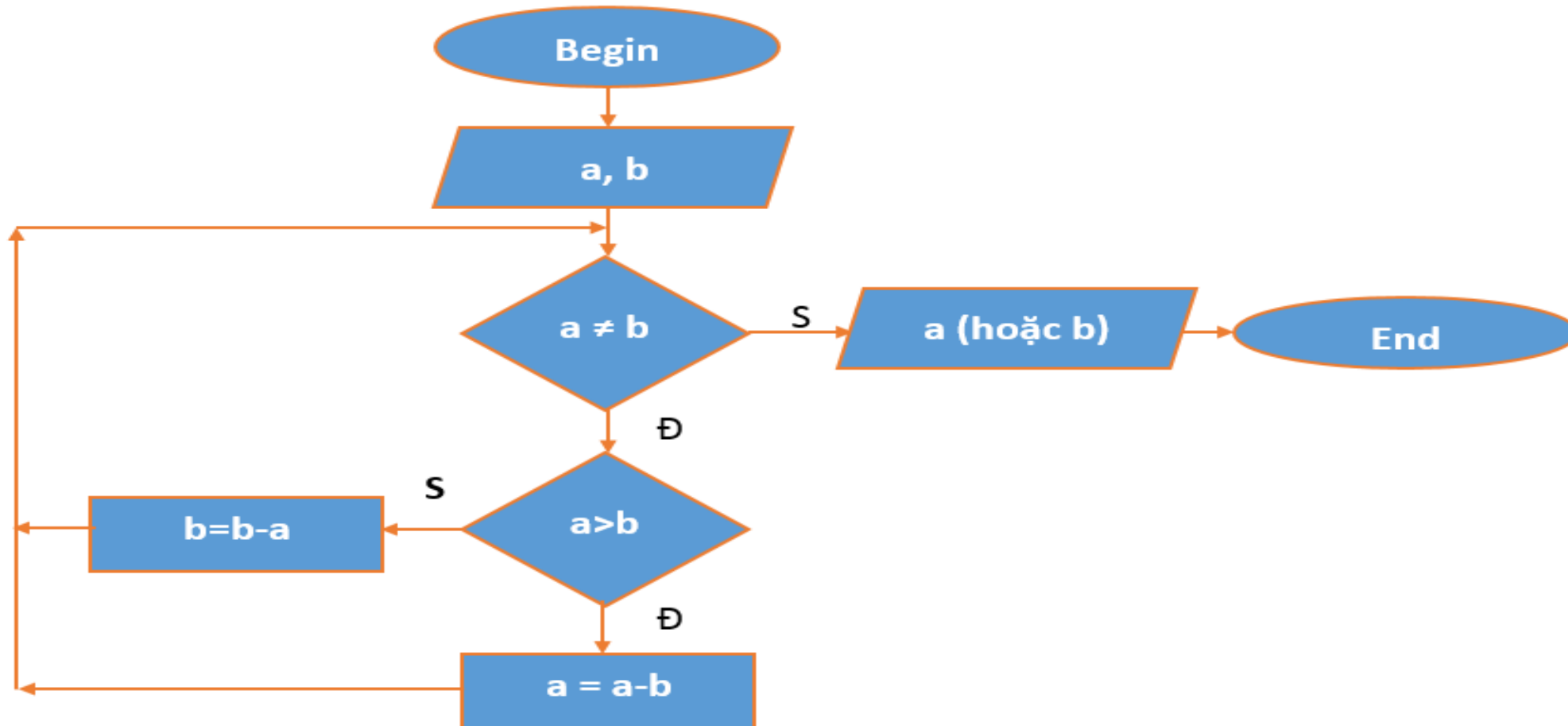


# Các khối cơ bản của một sơ đồ thuật toán

Biểu tượng	Ý nghĩa
	Bắt đầu thuật toán
	Kết thúc thuật toán
	Tính toán biểu thức
	Nhập, xuất dữ liệu
	Kiểm tra điều kiện để rẽ nhánh
	Hướng dữ liệu di chuyển

# Sơ đồ thuật toán tìm ƯSCLN

Bài toán: Tìm ước số chung lớn nhất của 2 số nguyên dương  $a, b$ .





## 1.2. Độ phức tạp thuật toán

\*

Tiêu chí đánh giá độ phức tạp thuật toán-2 tiêu chí

1

Thuật toán đơn giản, dễ hiểu, dễ cài đặt.

2

*Tài nguyên mà thuật toán sử dụng khi thực hiện*



# *Ví dụ - tìm USCLN*

- ❖ Cách 1. Tìm UCLN sử dụng phép trừ
- ❖ Cách 2. Tìm UCLN sử dụng phép chia dư
- ❖ Cách 3. Tìm UCLN sử dụng giải thuật Euclid
- ❖ Cách 4. Tìm UCLN sử dụng hàm có sẵn của C++



## Cách 2

```
❖ int gcd(int a, int b){  
❖     // Lặp tới khi 1 trong 2 số bằng 0  
❖     while (a*b != 0){  
❖         if (a > b){  
❖             a %= b; // a = a % b  
❖         }else{  
❖             b %= a;  
❖         }  
❖     }  
❖     return a + b;  
❖     // return a + b, bởi vì lúc  
❖     này hoặc a hoặc b đã  
❖     bằng 0.  
❖ }
```



# Đánh giá thời gian thực hiện thuật toán



Có 2 cách tiếp cận.....



Phương pháp thực nghiệm.....



*Phương pháp lý thuyết*.....



# Đánh giá thời gian thực hiện thuật toán



Phương pháp thực nghiệm:

Thời gian thực hiện chương trình phụ thuộc vào các yếu tố sau:



Dữ liệu vào



Chương trình dịch để chuyển mã nguồn thành mã máy



Tốc độ thực hiện của máy tính được sử dụng



# Đánh giá thời gian thực hiện thuật toán

## ❖ Phương pháp lý thuyết

- Thời gian thực hiện thuật toán được coi là một hàm số của kích thước dữ liệu vào
- Sử dụng hàm  $T(n)$  để biểu diễn thời gian thực hiện thuật toán
  - $n$  là kích thước dữ liệu vào
- $T(n)$  được tính là số phép toán sơ cấp cần tiến hành khi thực hiện thuật toán
  - Các phép toán sơ cấp:  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $>$ ,  $<$ ,  $>=$ ,.....
- $T(n)$  phụ thuộc vào kích thước và tình trạng dữ liệu vào
- $T(n)$  được đánh giá trong 3 trường hợp:
  - Tốt nhất
  - Xấu nhất
  - Trung bình





# Đánh giá thời gian thực hiện thuật toán

## ❖ Ký hiệu O (O lớn)

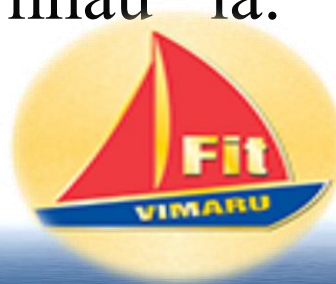
- Cho  $n$  là số nguyên không âm,  $T(n)$  và  $f(n)$  là các hàm thực không âm.
- Ta nói  $T(n)=O(f(n))$  nếu và chỉ nếu tồn tại các hằng số  $C, n_0$  sao cho  $T(n) \leq C.f(n)$  với mọi  $n \geq n_0$



# Đánh giá thời gian thực hiện thuật toán

❖ Quy tắc đánh giá: Cho P1, P2 là 2 đoạn chương trình có thời gian thực hiện lần lượt là:  $T1(n)=O(f1(n))$  và  $T2(n)=O(f2(n))$ .

- Quy tắc tổng: Thời gian thực hiện tuần tự P1 rồi P2 là  $T(n)=T1(n)+T2(n)=O(\max\{f1(n), f2(n)\})$
- Quy tắc nhân: Thời gian thực hiện P1 và P2 lồng nhau là:  $T(n)=T1(n)*T2(n)=O(f1(n)*f2(n))$



# Đánh giá thời gian thực hiện thuật toán

- ❖ Ví dụ 1: đoạn chương trình tìm x trong mảng gồm n phần tử  
for(i=0; i<n; i++)  
    if(a[i]==x) return i;  
return -1;
- ✓ Lệnh if thực hiện 1 phép toán cơ sở (so sánh ==)
- ✓ Lệnh for thực hiện n lần
- ✓ Do lệnh for và if lồng nhau nên áp dụng quy tắc nhân ta có thời gian thực hiện đoạn chương trình trên là  $1.n = n$



# *Đánh giá thời gian thực hiện thuật toán*

❖ Ví dụ 2: đoạn chương trình tìm số lớn nhất trong ma trận  $n$  hàng,  $m$  cột.

```
max=a[0][0];  
for(i=0; i<n; i++)  
for(j=0; j<m; j++)  
    if(a[i][j]>max) max=a[i][j];
```

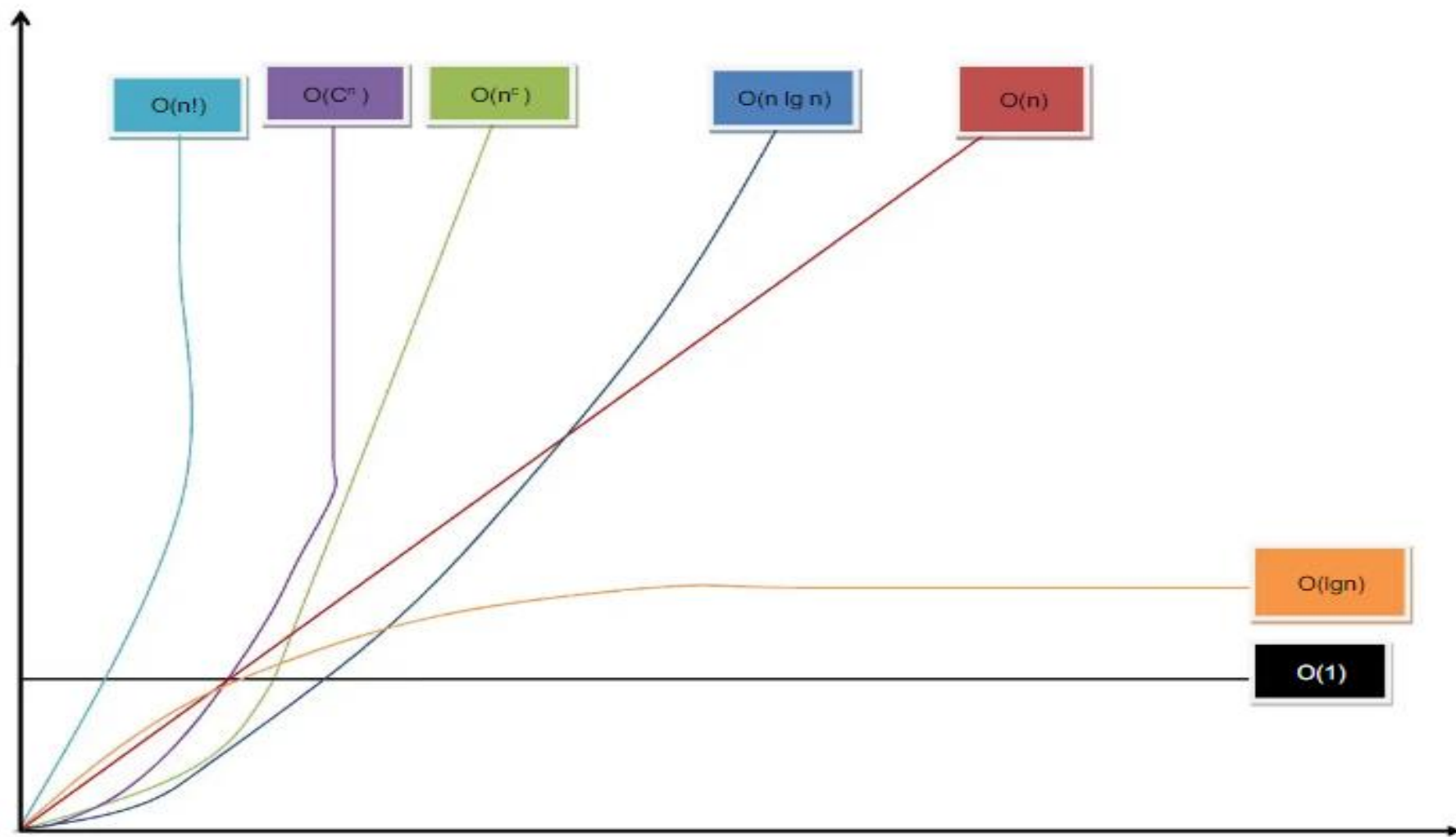
- ✓ Lệnh if thực hiện 1 phép toán cơ sở (so sánh  $>$ ).
- ✓ Lệnh for thứ nhất thực hiện  $n$  lần, for thứ 2 thực hiện  $m$  lần.
- ✓ Do lệnh for và if lồng nhau nên áp dụng quy tắc nhân ta có thời gian thực hiện đoạn chương trình trên là  $1.n.m=n.m$ .



# Phân lớp các hàm

Dạng O	Tên Phân loại	
$O(1)$	Hằng	
$O(\log_2(n))$	logarit	
$O(\sqrt{n})$	Căn thức	
$O(\sqrt[3]{n})$		
...		
$O(\sqrt[n]{n})$		
$O(n)$	Tuyến tính	Đa thức
$O(n^2)$	Bình phương	
$O(n^3)$	Bậc ba	
...		
$O(n^m)$	Đa thức	Độ phức tạp lớn
$O(c^n)$ , với $c > 1$	Mũ	
$O(n!)$	Giai thừa	

# *Phân lớp các thuật toán*





## *1.3. Cấu trúc dữ liệu (data struct)*

- ❖ Trong khoa học máy tính, **cấu trúc dữ liệu** là một cách lưu dữ liệu trong máy tính sao cho nó có thể được sử dụng một cách hiệu quả.
- ❖ Cấu trúc dữ liệu là cách lưu trữ, tổ chức dữ liệu có thứ tự, có hệ thống để dữ liệu có thể được sử dụng một cách hiệu quả.



## *1.3. Cấu trúc dữ liệu (tiếp)*

### ❖ Thành phần của ctdl:

- **Interface:** biểu diễn một tập hợp các phép tính mà một cấu trúc dữ liệu hỗ trợ.
- **Implementation** (có thể hiểu là sự triển khai): Cung cấp sự biểu diễn nội bộ của một cấu trúc dữ liệu.

❖ Các ctdl được xây dựng từ các kiểu dữ liệu cơ bản.



## 1.3. Cấu trúc dữ liệu (tiếp)

❖ Trong lĩnh vực khoa học máy tính, kiểu dữ liệu cơ bản (primitive data type) là kiểu được định nghĩa bởi một ngôn ngữ lập trình làm gốc để xây dựng các kiểu dữ liệu phức hợp khác.

- logic (lưu một trong hai giá trị true/false)
- ký tự (char: lưu giá trị ký tự)
- số thực (float, double: lưu giá trị số thực)
- số nguyên (int, long: lưu giá trị nguyên)
- kiểu liệt kê
- void



## *1.3. Cấu trúc dữ liệu (tiếp)*

- ❖ Một số ctdl hay dùng
  - Mảng
  - Ngăn xếp(stack)
  - Hàng đợi(queue)
  - Bảng băm(hash table)
  - Danh sách liên kết (linked list)
  - Cây (tree)
  - Đồ thị (graph)



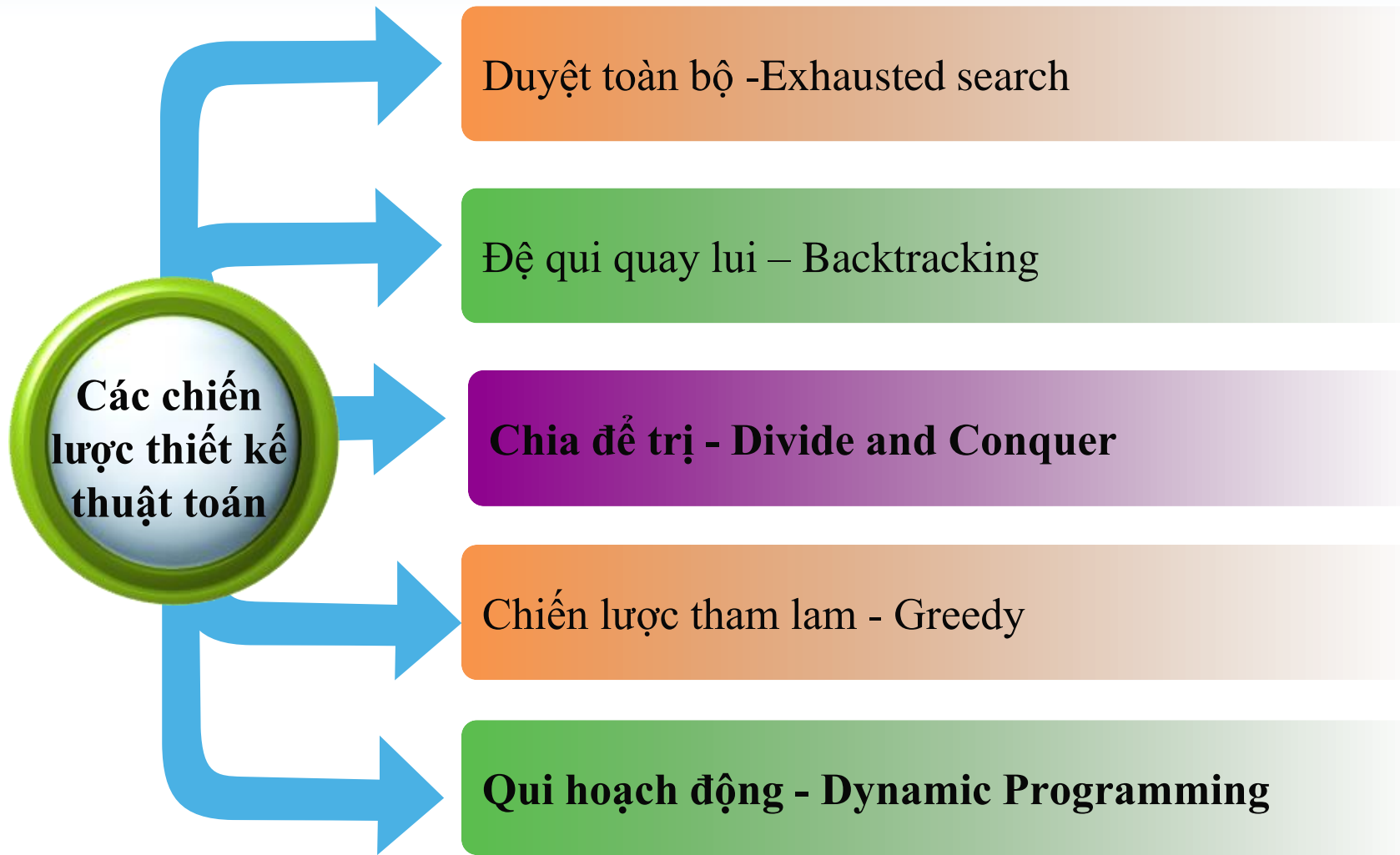
## *1.4. Mối quan hệ giữa ctdl và thuật toán*

❖ Chương trình (Programs) = Cấu trúc dữ liệu (Data Structures) + Giải thuật (Algorithms)

- Cấu trúc dữ liệu được sử dụng để biểu diễn dữ liệu
- Giải thuật được sử dụng để thực hiện các thao tác trên các dữ liệu của bài toán nhằm hoàn thành các chức năng của chương trình



# 1.5. Các chiến lược thiết kế thuật toán





# *Bài tập chương 1*

1. Vẽ sơ đồ khối thuật toán giải pt  $ax^2 + bx + c = 0$
2. Vẽ sơ đồ khối giải thuật kiểm tra một số nguyên dương có là số nguyên tố không? Đánh giá độ phức tạp của giải thuật sử dụng.
3. Vẽ sơ đồ khối giải thuật tìm số lớn nhất trong mảng n số nguyên. Đánh giá giải thuật sử dụng.
4. Vẽ sơ đồ khối giải thuật tìm bội số chung nhỏ nhất của 2 số nguyên dương. Đánh giá giải thuật sử dụng.
5. ...

