

# CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

## CHƯƠNG 1 TỔNG QUAN

GV: Bùi Văn Thượng  
Email: thuong.bui@ut.edu.vn

# NỘI DUNG CHÍNH

Mô hình  
hóa bài  
toán thực  
tế

Cấu trúc  
dữ liệu

Giải thuật

Chương  
trình

# MÔ HÌNH HÓA BÀI TOÁN THỰC TẾ

Mô hình hóa bài toán tức là nhận biết và giải quyết các bài toán thực tế theo hướng tin học hóa. Thông thường, các bài toán ban đầu thường không rõ ràng. Để giảm bớt sự phức tạp của bài toán thực tế, ta phải hình thức hóa nó, nghĩa là phát biểu lại bài toán thực tế thành một bài toán hình thức hay còn gọi là mô hình toán.



# MÔ HÌNH HÓA BÀI TOÁN THỰC TẾ

Sau khi đã xác định được mô hình toán học cho vấn đề cần giải quyết, tiếp theo, ta có thể hình thành một thuật toán cho mô hình đó.

Đầu tiên thuật toán thường được diễn tả dưới dạng các phát biểu mang tính tổng quát, và sau đó sẽ được tinh chỉnh dần từng bước thành chuỗi các lệnh cụ thể, rõ ràng hơn.

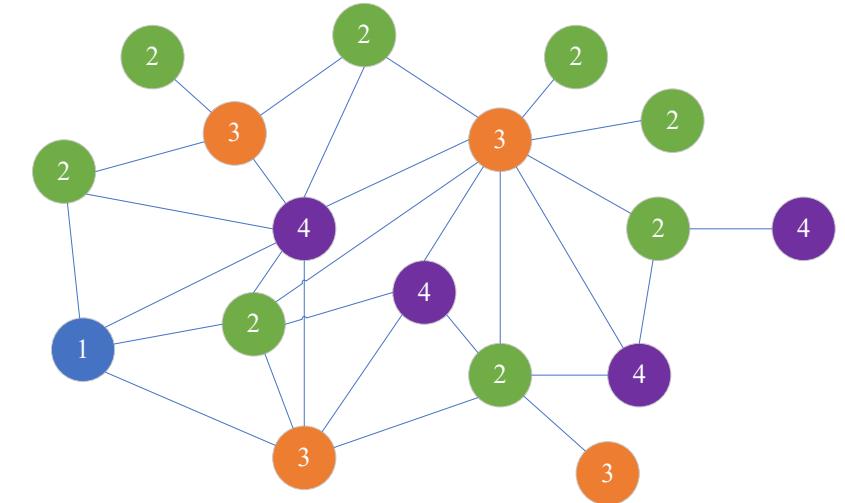
## Ví dụ: Bài toán tô màu bản đồ thế giới



Mỗi đỉnh đều phải được tô màu.

Hai đỉnh có cạnh nối thì phải tô bằng hai màu khác nhau.

Tô màu sao cho số màu được sử dụng là ít nhất.



# CẤU TRÚC DỮ LIỆU

**Cấu trúc dữ liệu** (tiếng Anh là Data Structure - sau đây sẽ viết tắt là CTDL) là cách tổ chức và lưu trữ dữ liệu trong máy tính để có thể truy xuất và xử lý dễ dàng. Nó giúp tăng hiệu suất và hiệu quả của chương trình bằng cách cung cấp các phương pháp tiếp cận và thao tác dữ liệu hiệu quả.

**Kiểu dữ liệu** là một tập hợp các giá trị và một tập hợp các phép toán trên các giá trị đó.

# CẤU TRÚC DỮ LIỆU

Kiểu dữ liệu có hai loại là kiểu dữ liệu sơ cấp và kiểu dữ liệu có cấu trúc:

- ☞ **Kiểu dữ liệu sơ cấp** là kiểu dữ liệu mà giá trị dữ liệu của nó là đơn nhất. Chẳng hạn ta có một số kiểu dữ liệu sơ cấp như: boolean, integer, character, float, ....
- ☞ **Kiểu dữ liệu có cấu trúc** hay còn gọi là cấu trúc dữ liệu là kiểu dữ liệu mà giá trị dữ liệu của nó là sự kết hợp của các giá trị khác. Chẳng hạn ta có một số cấu trúc dữ liệu như: mảng, kiểu cấu trúc, danh sách liên kết, ....

# CẤU TRÚC DỮ LIỆU

## Tiêu chuẩn lựa chọn cấu trúc dữ liệu

Cấu trúc dữ liệu đóng vai trò quan trọng trong việc kết hợp và đưa ra cách giải quyết cho bài toán. Nó hỗ trợ cho các thuật toán thao tác trên đối tượng được hiệu quả hơn.

Việc lựa chọn cấu trúc dữ liệu cho bài toán có thể dựa vào các tiêu chuẩn sau:

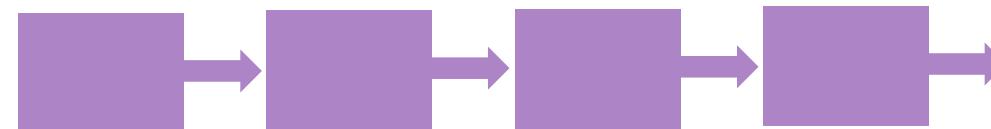
- ☞ *Phải biểu diễn được đầy đủ thông tin nhập và xuất của bài toán.*
- ☞ *Phải phù hợp với các thao tác của thuật toán mà ta lựa chọn.*
- ☞ *Phù hợp với điều kiện cho phép của ngôn ngữ lập trình đang sử dụng.*
- ☞ *Tiết kiệm tài nguyên hệ thống.*

# CẤU TRÚC DỮ LIỆU

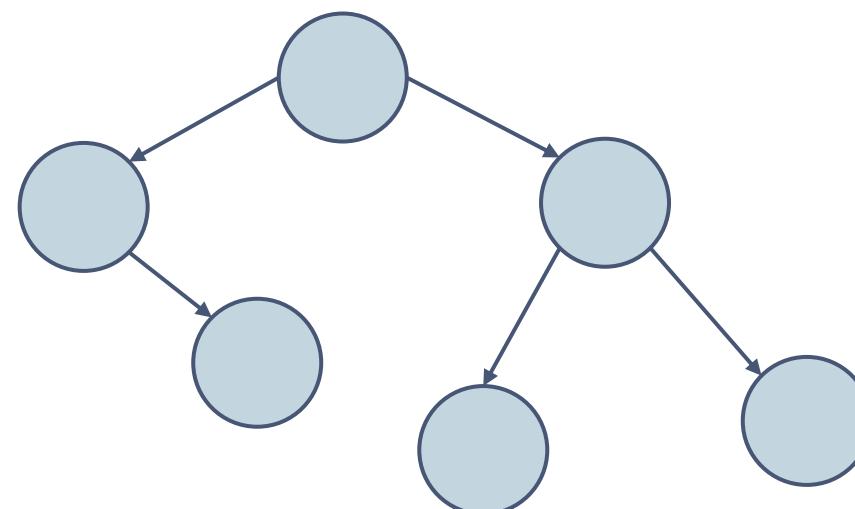
## Các kiểu cấu trúc dữ liệu

Cấu trúc dữ liệu có thể chia thành các dạng phổ biến sau:

**1. Cấu trúc dữ liệu tuyến tính:** Là *cấu trúc dữ liệu* trong đó các phần tử được *liên kết tuần tự, nối tiếp với nhau* (ví dụ: *Mảng, danh sách liên kết, hàng đợi, ...*).



**2. Cấu trúc dữ liệu dạng cây:** Mỗi phần tử có thể liên kết với nhiều phần tử khác theo từng mức.

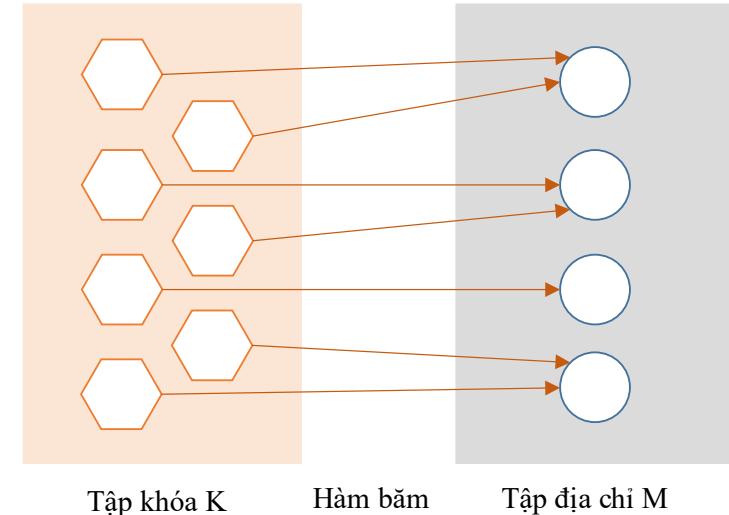


# CẤU TRÚC DỮ LIỆU

## Các kiểu cấu trúc dữ liệu

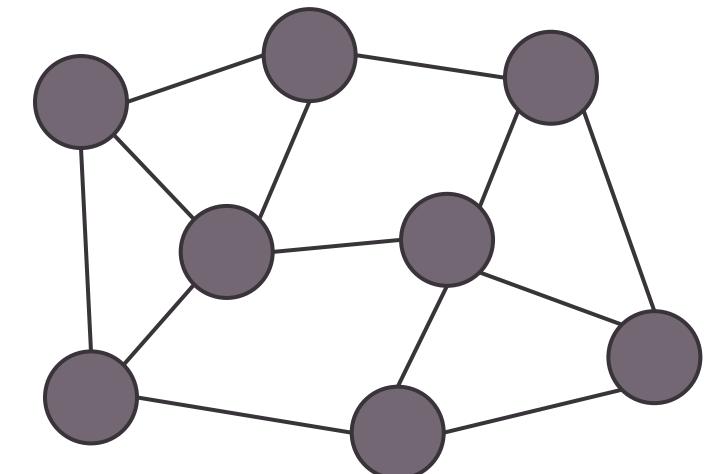
### 3. Cấu trúc dữ liệu bảng băm:

Là một cấu trúc dữ liệu tương tự như mảng nhưng kèm theo một hàm băm để ánh xạ nhiều giá trị vào cùng một phần tử trong mảng.



### 4. Cấu trúc dữ liệu dạng đồ thị:

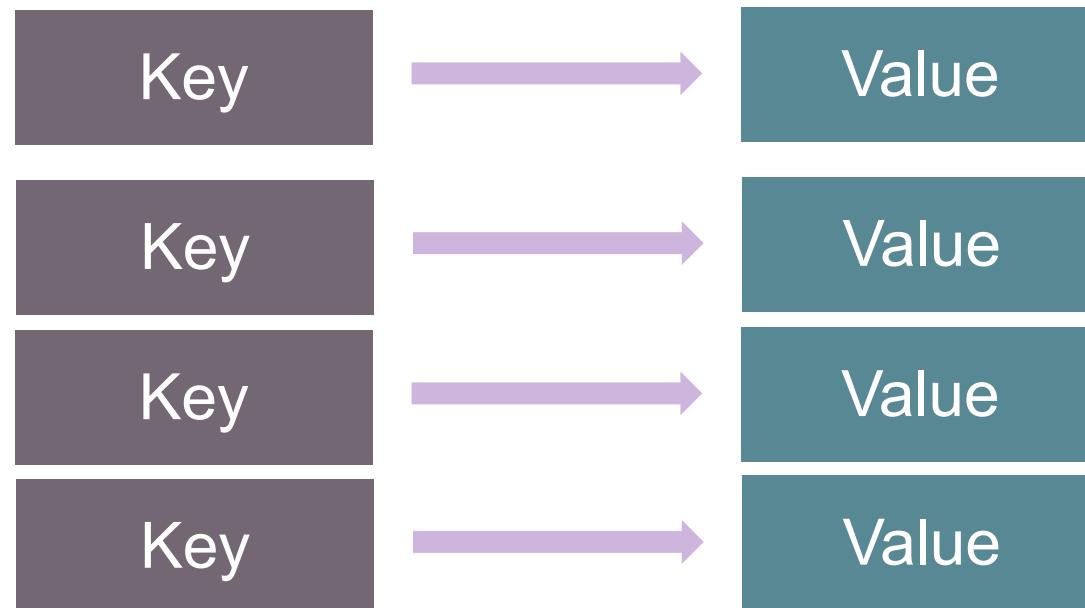
Là dạng cấu trúc dữ liệu bao gồm một tập các đối tượng được gọi là các đỉnh (hoặc nút) nối với nhau bởi các cạnh (hoặc cung).



# CẤU TRÚC DỮ LIỆU

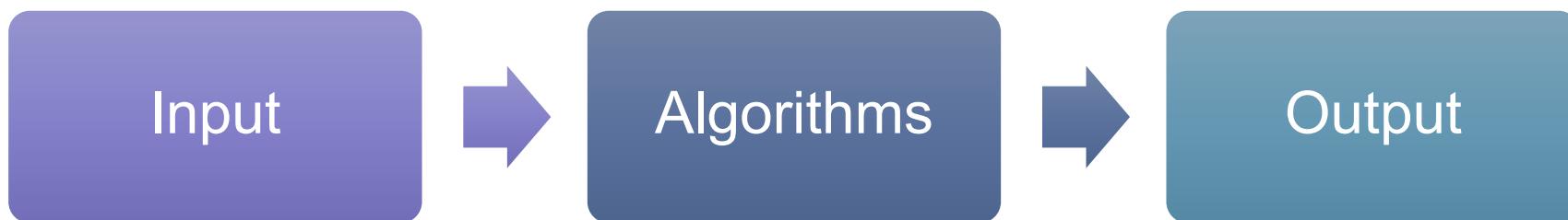
## Các kiểu cấu trúc dữ liệu

**5. Cấu trúc dữ liệu ánh xạ:** Là cấu trúc dữ liệu mà mỗi phần tử là ánh xạ giữa khóa (key) và giá trị (value) của nó (ví dụ: map, JSON, ...).



# GIẢI THUẬT

Giải thuật (hay còn gọi là thuật toán - tiếng Anh là Algorithms) là một chuỗi hữu hạn các thao tác để giải một bài toán nào đó. Giải thuật là độc lập với các ngôn ngữ lập trình, tức là một giải thuật có thể được triển khai trong nhiều ngôn ngữ lập trình khác nhau.



# GIẢI THUẬT

Các tính chất quan trọng của giải thuật bao gồm:

- ☞ **Tính hữu hạn:** Giải thuật phải luôn kết thúc sau một số hữu hạn bước.
- ☞ **Tính xác định:** Mỗi bước của giải thuật phải được xác định rõ ràng, phải được thực hiện chính xác, và phải mang một mục đích nhất định.
- ☞ **Tính đúng:** Giải thuật phải đảm bảo tính đúng và chính xác.
- ☞ **Tính phổ biến:** Giải thuật phải giải quyết được lớp các vấn đề tương tự.
- ☞ **Tính hiệu quả:** Các thao tác trong giải thuật phải có khả năng giải quyết hiệu quả vấn đề trong điều kiện thời gian và tài nguyên cho phép.
- ☞ **Dữ liệu đầu vào xác định (input):** Giải thuật phải xử lý một số lượng dữ liệu đầu vào xác định.
- ☞ **Kết quả đầu ra (output):** Giải thuật sẽ giải quyết các bài toán bằng cách xử lý các loại dữ liệu đầu vào để cho kết quả dữ liệu đầu ra đã xác định

# GIẢI THUẬT

Ví dụ: Thuật toán để giải phương trình  $P(x)$ :  $ax + b = 0$ .

Input:  $a, b$

Output: Nghiệm  $x$

Thực hiện:

- Nếu  $a = 0 \implies$  xét  $b$ :
  - Nếu  $b = 0$  thì  $P(x)$  có nghiệm bất kì
  - Nếu  $b \neq 0$  thì  $P(x)$  vô nghiệm
- Nếu  $a \neq 0$  thì  $P(x)$  có duy nhất một nghiệm  $x = (-b)/a$

# GIẢI THUẬT

**Các cách thường được sử dụng để biểu diễn giải thuật:**

1. **Sử dụng ngôn ngữ tự nhiên:** *Liệt kê tuần tự các bước để giải quyết bài toán. Cách này đơn giản và không cần kiến thức về biểu diễn giải thuật, tuy nhiên nó thường dài dòng và đôi khi khó hiểu.*

**Ví dụ:** Giải thuật tìm số lớn nhất trong 3 số a, b, c.

Input: a, b, c.

Output: max (a, b, c).

Biểu diễn giải thuật bằng ngôn ngữ tự nhiên:

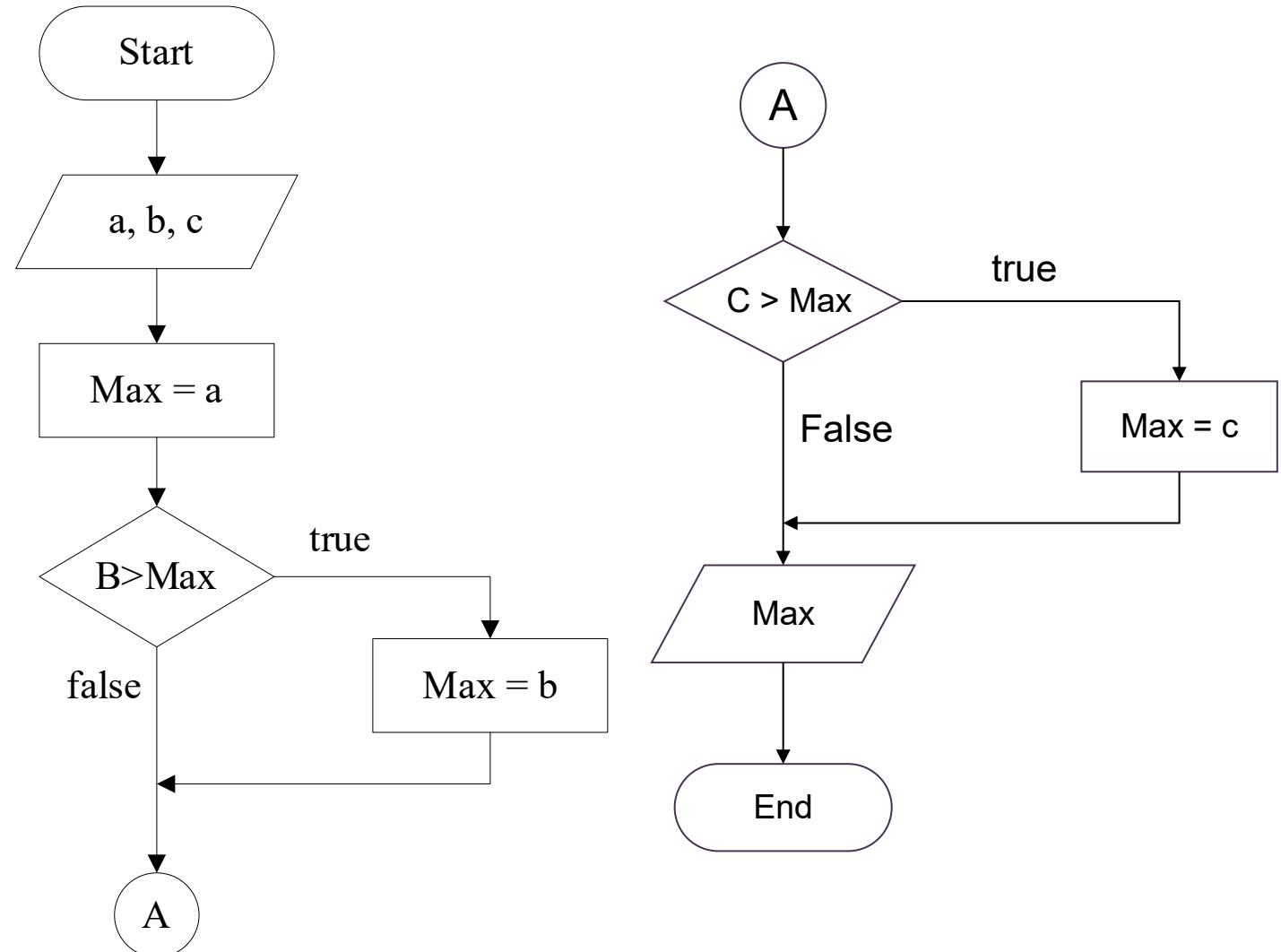
- *Đặt biến max = a.*
- *Nếu max < b ta gán lại max = b.*
- *Nếu max < c, ta gán lại max = c.*
- *Số lớn nhất cuối cùng tìm được trong 3 số a, b, c có giá trị bằng max.*

# GIẢI THUẬT

## 2. Sử dụng lưu đồ (sơ đồ khối):

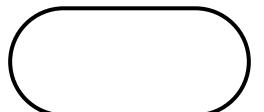
Sử dụng các hình khối khác nhau để biểu diễn giải thuật. Cách này trực quan và dễ hiểu, tuy nhiên hơi cồng kềnh.

**Ví dụ:** Lưu đồ giải thuật tìm giá trị lớn nhất trong 3 số a, b, c (hình bên).

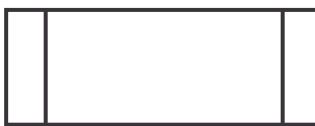


# GIẢI THUẬT

## Quy tắc biểu diễn giải thuật bằng sơ đồ khối



Bắt đầu hoặc kết thúc



Khối chương trình con



Input, output



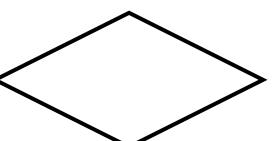
Hướng đi của thuật toán



Quá trình xử lý, tính toán



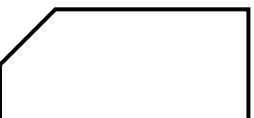
Thao tác chuẩn bị



Điều kiện rẽ nhánh



Điểm nối



Tập tin dữ liệu



Chú thích

# GIẢI THUẬT

**3. Sử dụng mã giả:** Sử dụng ngôn ngữ tựa ngôn ngữ lập trình để biểu diễn giải thuật. Cách làm này đỡ cồng kềnh hơn so với đồ khối tuy nhiên không trực quan.

**Ví dụ:** Giải thuật tìm số lớn nhất trong 3 số a, b, c.

Input:  $a, b, c$

Output: Số lớn nhất  $max\_val$

**Mã giả:**

$max\_val = a$

*IF*  $b > max\_val$

*THEN*

$max\_val = b$

*IF*  $c > max\_val$

*THEN*

$max\_val = c$

*RETURN*  $max\_val$

# GIẢI THUẬT

**4. Sử dụng ngôn ngữ lập trình:** Sử dụng các ngôn ngữ máy tính như C, Python, ... để biểu diễn giải thuật. Cách này đòi hỏi phải có kiến thức và kỹ năng về ngôn ngữ lập trình được sử dụng.

**Ví dụ:** Tìm số lớn nhất trong 3 số a, b, c.

```
#include<iostream>
using namespace std;
int main(){
    int a,b,c,max;
    cout<<"Nhập 3 số bất kỳ: ";
    cin>>a>>b>>c;
    max = a;
    if(b>max) max=b;
    if(c>max) max=c;
    cout<<"Số lớn nhất là: "<<max<<endl;
    system("pause");
}
```

# ĐỘ PHÚC TẠP CỦA GIẢI THUẬT

## Đánh giá độ phức tạp của thuật toán

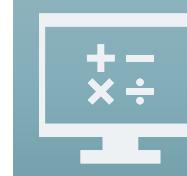
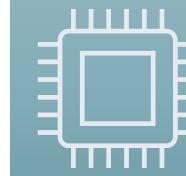
Với phần lớn các bài toán, thường có nhiều giải thuật khác nhau để giải. Để so sánh và lựa chọn giải thuật tốt nhất người ta thường phân tích và dự đoán các tài nguyên mà giải thuật đó cần.

Một giải thuật được gọi là hiệu quả nếu chi phí cần sử dụng tài nguyên của máy là thấp. Trong đó, thời gian tính toán là tài nguyên quan trọng nhất



# ĐỘ PHÚC TẠP CỦA GIẢI THUẬT

## Các yếu tố ảnh hưởng tới thời gian chạy của giải thuật



# ĐỘ PHÚC TẠP CỦA GIẢI THUẬT

Các phương pháp đánh giá độ phức tạp của giải thuật:

## 1. Phương pháp thực nghiệm:

- Cách làm: Cài giải thuật rồi chọn các bộ dữ liệu thử nghiệm, sau đó thống kê các thông số nhận được khi chạy các bộ dữ liệu đó để đánh giá.
- Ưu điểm: Dễ thực hiện.
- Nhược điểm:
  - Chịu sự hạn chế của ngôn ngữ lập trình.
  - Ảnh hưởng bởi trình độ của người lập trình.
  - Khó khăn và tốn nhiều chi phí cho việc chọn các bộ dữ liệu thử đặc trưng cho tất cả tập các dữ liệu vào của thuật toán.
  - Phụ thuộc vào sức mạnh của phần cứng.

# ĐỘ PHÚC TẠP CỦA GIẢI THUẬT

## 2. Phương pháp xấp xỉ toán học:

- Đánh giá thuật toán theo hướng xấp xỉ tiệm cận qua các khái niệm Big-O.
- Ưu điểm: Ít phụ thuộc môi trường cũng như phần cứng hơn.
- Nhược điểm: Phức tạp.

Thời gian tính toán của một giải thuật thể hiện số phép toán của giải thuật so với kích thước của dữ liệu đầu vào. Độ phức tạp của giải thuật được mô tả qua một hàm  $f(N)$ , trong đó  $N$  là khối lượng dữ liệu cần được xử lý. Các trường hợp độ phức tạp quan tâm:

- Trường hợp tốt nhất (phân tích chính xác).
- Trường hợp xấu nhất (phân tích chính xác).
- Trường hợp trung bình (mang tính dự đoán).

# ĐỘ PHÚC TẠP CỦA GIẢI THUẬT

## 2. Phương pháp xấp xỉ toán học:

Độ phức tạp của thuật toán được đánh giá qua một hàm có ký hiệu là Big-O. Big-O có tham số kích thước dữ liệu đầu vào là  $n$  và nó thể hiện thời gian chạy lâu nhất của một thuật toán (trường hợp xấu nhất).

**Ví dụ:** Độ phức tạp của giải thuật tính tổng  $1 + 2 + \dots + n$ .

- Bài này nếu dùng công thức để tính:  $n*(n+1)/2$ . Giải thuật này có độ phức tạp là  $O(1)$  (1 phép toán).
- Nếu dùng vòng lặp từ 1 đến  $n$  để tính tổng, độ phức tạp là  $O(n)$ . Với  $n$  bằng 1 tỷ, tương đương với việc thực hiện 1 tỷ lần phép toán cộng.

# ĐỘ PHÚC TẠP CỦA GIẢI THUẬT

Bảng phân mức độ phức tạp của thuật toán:

Ký hiệu Big-O	Độ phức tạp	Mô tả
$O(c)$	Hằng số	Số phép tính là hằng số c, không phụ thuộc vào khối lượng dữ liệu đầu vào.
$O(\log(n))$	Logarit	Số phép tính tăng theo hàm logarit.
$O(n)$	Tuyến tính	Số phép tính tỷ lệ thuận với dữ liệu đầu vào.
$O(n(\log(n)))$	Tuyến tính logarit	Tổng hợp của n và $\log(n)$ . Ví dụ: Heap sort, Merge sort, ...
$O(n^2)$	Lũy thừa bậc 2	Số phép toán là đa thức lũy thừa bậc $n^2$ .
$O(n^3)$	Lũy thừa bậc 3	Số phép toán là đa thức lũy thừa bậc $n^3$ .
$O(2^n)$	Hàm mũ	Số phép toán là hàm mũ $2^n$ .
$O(n!)$	Giai thừa	Độ phức tạp giai thừa.

# CHƯƠNG TRÌNH

**Chương trình** là một hình thức khác của thuật toán, được viết bằng một ngôn ngữ lập trình để máy tính có thể hiểu và thực hiện. Dựa trên dữ liệu đầu vào, chương trình tiến hành xử lý và trả về kết quả đầu ra.

Chương trình máy tính tồn tại dưới hai dạng: chương trình có thể chạy được và mã nguồn chương trình. Khi kết hợp một hoặc nhiều chương trình với dữ liệu liên quan, chúng ta có **phần mềm máy tính**.



# CHƯƠNG TRÌNH

Một chương trình máy tính cần đảm bảo các tiêu chuẩn sau:

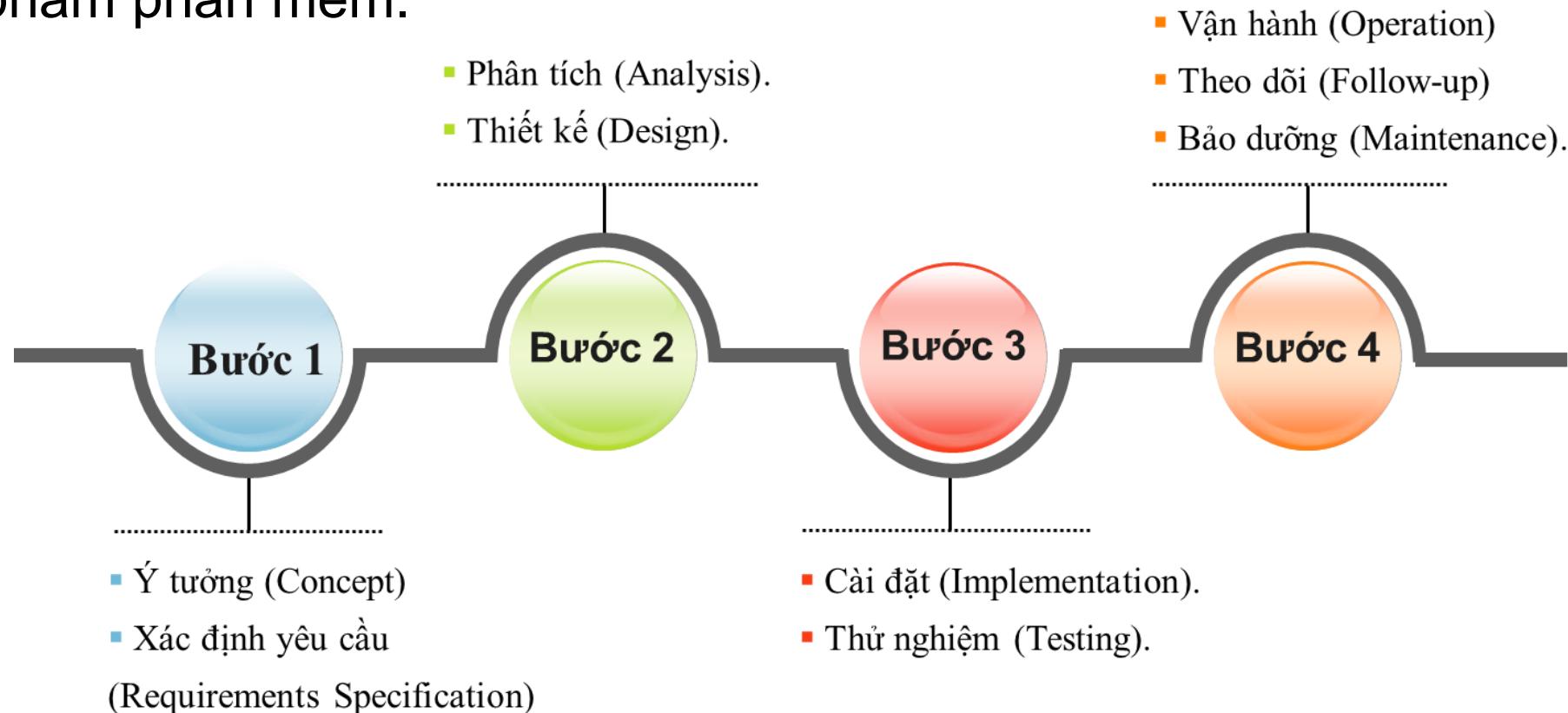
- **Tính tin cậy:** *Phải chạy đúng như dự định, mô tả chính xác một giải thuật đúng.*
- **Tính uyển chuyển:** *Dễ sửa đổi, mở rộng, đáp ứng các quy trình làm phần mềm.*
- **Tính trong sáng:** *Mã nguồn dễ đọc, dễ hiểu. Tính trong sáng phụ thuộc rất nhiều vào công cụ lập trình và phong cách lập trình.*
- **Tính hữu hiệu:** *Đáp ứng được yêu cầu là chạy nhanh và ít tốn tài nguyên bộ nhớ, điều này phụ thuộc vào chất lượng của giải thuật cũng như kỹ năng của lập trình viên.*

Tù những phân tích trên ta thấy rằng việc tạo ra một chương trình đòi hỏi rất nhiều công đoạn và tiêu tốn rất nhiều công sức. Vì vậy đừng bao giờ viết chương trình mà chưa suy xét kỹ về giải thuật và những dữ liệu cần thao tác.

# CHƯƠNG TRÌNH

## Quy trình làm phần mềm

Quy trình phát triển phần mềm là một cấu trúc bao gồm tập hợp các thao tác và các kết quả tương quan sử dụng trong việc phát triển để sản xuất ra một sản phẩm phần mềm.



# BÀI TẬP CHƯƠNG 1

**Câu 1.** Trình bày các tiêu chuẩn lựa chọn cấu trúc dữ liệu?

**Câu 2.** Hãy vẽ lưu đồ của giải thuật giải phương trình bậc hai:  $ax^2 + bx + c = 0$ .

**Câu 3.** Trình bày giải thuật tính tổng các chữ số của một số nguyên dương bất kỳ. Xác định độ phức tạp của giải thuật?

**Câu 4.** Hãy tìm hiểu và so sánh mô hình thác nước và mô hình chữ V cho việc xây dựng quy trình phát triển phần mềm?

**Câu 5.** Tìm hiểu mô hình phát triển phần mềm Agile và quy trình Scrum?

