

# PHÂN TÍCH VÀ THIẾT KẾ THUẬT TOÁN

cuu duong than cong. com

Phạm Thế Bảo

[ptbao@math.hcmuns.edu.vn](mailto:ptbao@math.hcmuns.edu.vn)

cuu duong than cong. com

<http://www.math.hcmuns.edu.vn/~ptbao/AlgorithmAnalysis/>

# Nội dung

- Tổng quan về thuật toán và độ phức tạp của thuật toán
- Đánh giá thuật toán bằng:
  - Công cụ toán học sơ cấp
  - Thực nghiệm
  - Hàm sinh
  - Hoán vị
- Đệ quy và phương pháp đánh giá
- Đánh giá một số thuật toán thông dụng
- Các phương pháp giải quyết bài toán trên máy tính:
  - Trực tiếp
  - Gián tiếp
- Kỹ thuật thiết kế thuật toán:
  - Chia để trị
  - Greedy
  - Quy hoạch động
  - Tìm kiếm cục bộ (địa phương)

# Hình thức kiểm tra

- Thực hành (4 điểm):
  - Làm việc theo nhóm
  - Mỗi nhóm sẽ đánh giá một thuật toán:
    - Chạy 20 loại bộ dữ liệu:  $50 \cdot i$  phần tử, với  $i=1..20$
    - Mỗi loại bộ dữ liệu chạy  $300 \cdot k$  lần, với  $k=1..10$
    - Mọi lần chạy dữ liệu được phát sinh ngẫu nhiên
  - Vẽ đồ thị, tính phương sai độ lệch chuẩn
  - Ước lượng độ phức tạp
  - Viết báo cáo
- Lý thuyết (6 điểm)

# Tài liệu tham khảo

1. Cẩm nang thuật toán – cuốn 1 – Robert Sedgewich – Trần Đan Thư.
2. Lập trình = Thuật toán + CTDL, N. Wirth
3. Algorithm Complexity & Communication Problems, J.P. Barthélemy, G. Cohen & a. Lobstein, UCL Press, London 1996.
4. Elementary Introduction to new Generalized Functions, Jean Francois Colombeau, 1991.
5. Algorithm and Complexity, Herbert S.Wilf, 1994.
6. Giải một bài toán trên máy tính như thế nào, Hoàng Kiêm, 2003.
7. The Art of Computer Vol. 1, 2, 3, Donald Knuth, Addison-Wesley

# Tổng quan về thuật toán

## 1. Thuật toán là gì?

Tập hợp hữu hạn các hướng dẫn rõ ràng để giải quyết một bài toán (vấn đề).

- Mở rộng (máy tính): một dãy **hữu hạn** các bước **không mập mờ** và **có thể thực thi được**, quá trình hành động theo các bước này phải **dừng** và cho được **kết quả như mong muốn**.

## 2. Tính chất cơ bản của thuật toán:

- Xác định = không mập mờ + thực thi được
- Hữu hạn
- Đúng

### 3. Ví dụ:

- Một lớp học cần chọn lớp trưởng theo các bước:
  1. Lập danh sách sinh viên
  2. Sắp thứ tự
  3. Chọn người đứng đầu làm lớp trưởng
- Danh sách cần gì?
- Sắp theo thứ tự nào? (tăng giảm, tiêu chí nào)
- Nếu trùng tiêu chí thì giải quyết ra sao?

## Sửa lại:

- a) Lập danh sách theo: họ tên, ngày tháng năm sinh, điểm các môn, điểm trung bình cuối năm.
  - b) Sắp xếp theo ĐTB giảm. Nếu ĐTB bằng nhau → cùng hạng.
  - c) Nếu có 01 HS đứng đầu → chọn, ngược lại chọn người có điểm toán cao nhất, nếu không chọn được → bốc thăm.
- Phân biệt mập mờ và lựa chọn có quyết định:
    - Mập mờ là thiếu thông tin hoặc có nhiều lựa chọn nhưng không đủ điều kiện quyết định, ví dụ: bước 1, 2.
    - Lựa chọn có quyết định là hoàn toàn xác định duy nhất trong điều kiện cụ thể của vấn đề, ví dụ bước c.

- Tính thực thi được, ví dụ:
  - Tính  $\sqrt{-1}$ ?
  - Chạy xe thẳng từ nhà hát lớn đến nhà thờ đức bà theo đường Đồng Khởi?
- Tính dừng, ví dụ:
  - B1:      nhập n;
  - B2:      s=0;
  - B3        i=1;
  - B4        nếu i=n+1 sang B8, ngược lại sang B5
  - B5        cộng i vào s
  - B6        cộng 2 vào i
  - B7        quay lại B4
  - B8        Tổng cần tính là s



- Đặc trưng khác của thuật toán:
  - Xác định đầu vào/ra
  - Tính hiệu quả: khối lượng tính toán, không gian, thời gian.
  - Tính tổng quát

Ví dụ:

- giải  $ax^2 + bx + c = 0$
  - Cho mảng các số nguyên A, tìm phần tử lớn nhất.
- Các phương pháp biểu diễn thuật toán:
  - Ngôn ngữ tự nhiên
  - Sơ đồ (lưu đồ) khối
  - Mã giả (Pseudo-code)

# Khái niệm thuật giải

## 1. Thuật giải là gì?

Các cách giải chấp nhận được nhưng không hoàn toàn đáp ứng đầy đủ các tiêu chuẩn của thuật toán thường được gọi là các thuật giải.

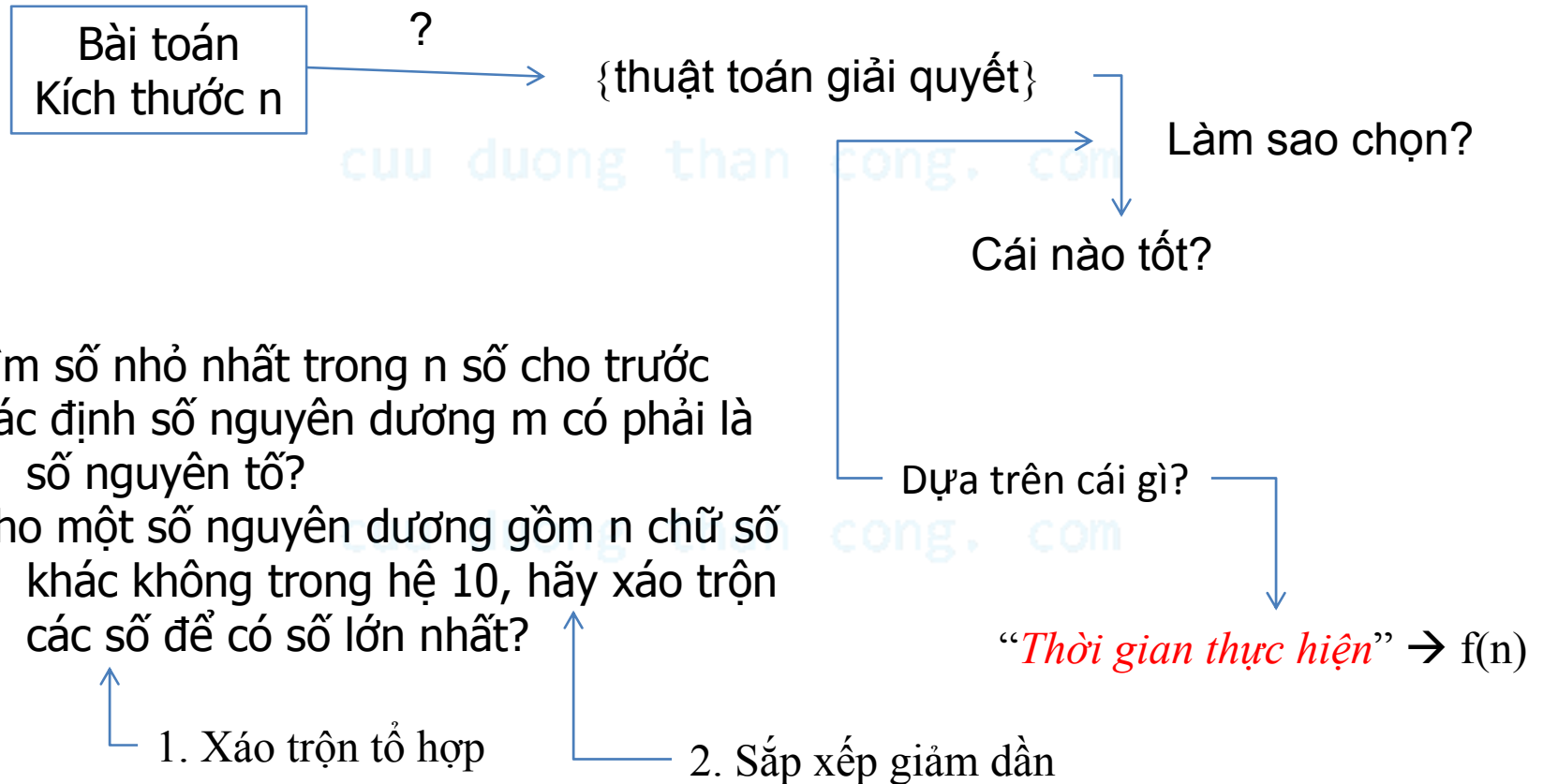
Đây là khái niệm mở rộng của thuật toán dựa trên tính xác định và tính đúng đắn.

Ví dụ thuật giải Heuristic:

- Nguyên lý vét cạn thông minh
- Nguyên lý Greedy (tham lam)
- Nguyên lý thử tự

# Độ phức tạp của thuật toán

## 1. Giới thiệu



Ví dụ:

- Tìm số nhỏ nhất trong n số cho trước
- Xác định số nguyên dương m có phải là số nguyên tố?
- Cho một số nguyên dương gồm n chữ số khác không trong hệ 10, hãy xáo trộn các số để có số lớn nhất?

Làm sao xác định “*thời gian thực hiện*”  $f(n)$ ?

## 1. Hướng tiệm cận:

- Lý thuyết
- Thực nghiệm

## 2. Công cụ toán học:

- Kỹ thuật sơ cấp
- Hàm sinh
- Hoán vị và nghịch thế

cuu duong than cong. com

# Hướng tiếp cận thực nghiệm

- Các bước thực hiện:
  1. Viết chương trình cài đặt
  2. Thực thi chương trình với nhiều bộ dữ liệu
  3. Đo và thống kê thời gian
  4. Xấp xỉ biểu đồ
- Hạn chế:
  1. Cần phải cài đặt CT và đo thời gian
  2. Bộ dữ liệu không thể đặc trưng hết
  3. Khó so sánh 02 thuật giải

# Ước lượng tiệm cận

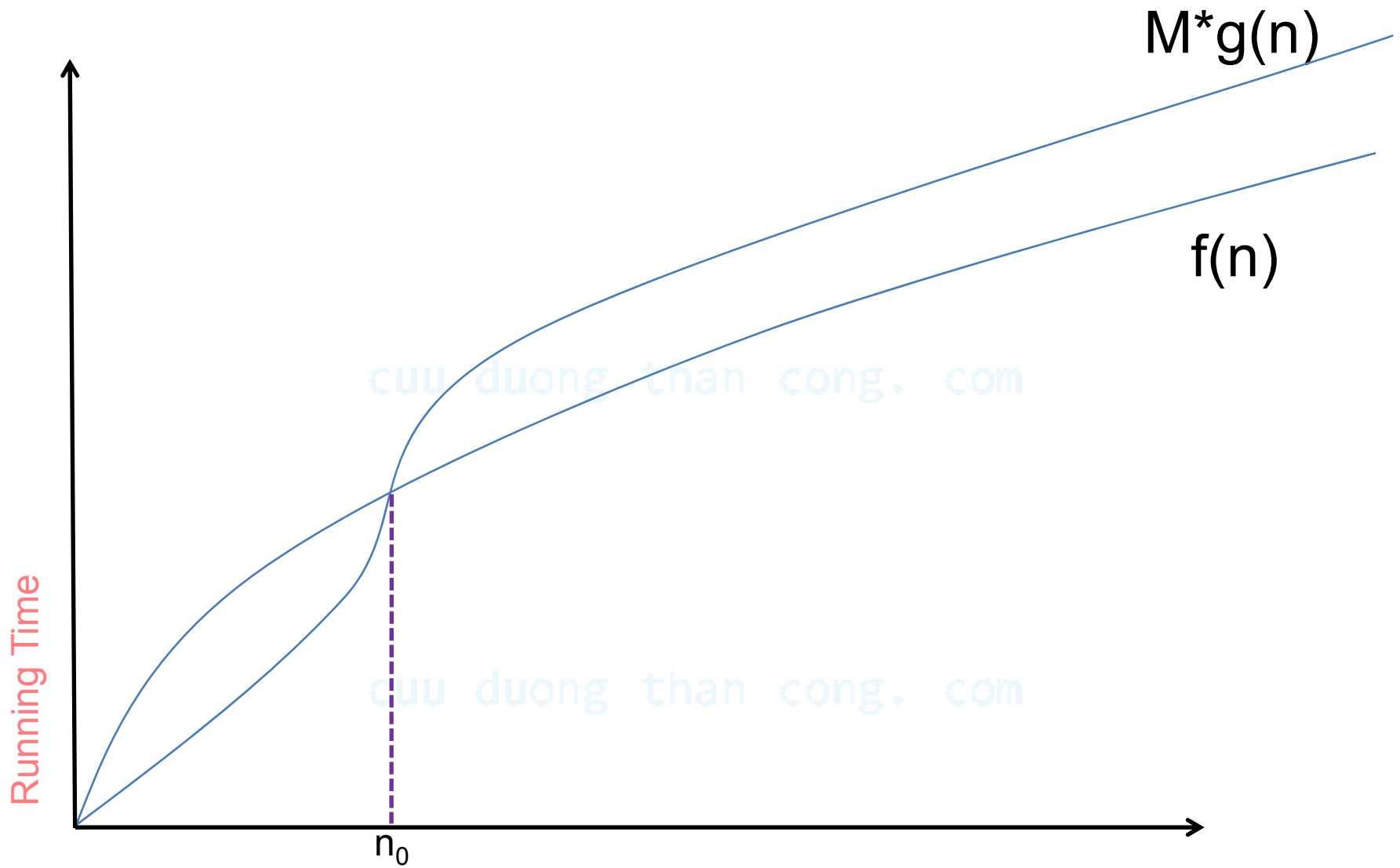
## 1. Ý nghĩa:

Phân lớp cấp độ lớn của các hàm  $f(n)$  khi  $n$  đủ lớn.

Ký hiệu  $O$  (big  $O$  –  $O$  lớn)

## 2. Định nghĩa:

Cho 2 hàm  $f, g : \mathbb{N} \rightarrow \mathbb{R}$ , ta nói  $f = O(g)$  nếu  
 $\exists n_0 \in \mathbb{N}$  và  $M > 0$ , sao cho  $|f(n)| \leq M |g(n)|$ ,  
 $\forall n \geq n_0$ .



# Ước lượng tiệm cận

- Mục đích:

Tìm  $f(n)$  được ước lượng dựa trên những hàm  $g(n)$  đã biết

Ví dụ:

$$1,000,001 \approx 1,000,000$$

$$3n^2 \approx n^2$$

- Ví dụ:

- Xem  $f(n)=n$  và  $g(n)=n^2$ , ta có  $f=O(g)$ , vì với  $M=1$  và  $n_0=1$ . Ta có  $|f(n)| \leq 1 \cdot |g(n)|, \forall n \geq 1$ .



- Xét  $f(n)=10000n$  và  $g(n)=n^2$  ta vẫn có  $f=O(g)$  vì
  - $|f(n)| \leq 10000 |g(n)|, \forall n \geq 1$
  - Hay  $|f(n)| \leq 1. |g(n)|, \forall n \geq 10000$
- Câu hỏi:  $g=O(f)$  ?  
 Giả sử  $g=O(f)$  thì có  $M$  và  $n_0$  sao cho  
 $n^2 \leq M (10000n), \forall n \geq n_0 \Rightarrow n \leq 10000 M ), \forall n \geq n_0$   
 $\Rightarrow$  vô lý.
- Xét  $f(n)=10n$  thì ta thấy
  - ❖  $f = O(n)$
  - ❖  $f = O(n^2)$
  - ❖  $f = O(n^3)$

- Cách viết khác:  $f \in O(g)$   
Ví dụ:  $10n \in O(n) \in O(n^2)$
- Tránh lý luận ngược biện:

$$\frac{1}{2} n^2 = O(n^2) = n^2 + 1$$

$$\Rightarrow \frac{1}{2} n^2 = n^2 + 1 \Rightarrow \text{S a i}$$

- Thực chất  $\frac{1}{2} n^2 \in O(n^2)$  và  $n^2 + 1 \in O(n^2)$

Tránh viết:  $O(n^2) = n^2 + 1$

Viết hợp lệ:

$$n^2 + 1 = O(n^2)$$

$$n^2 + 1 \in O(n^2)$$

- Thuật toán  $T$  có thời gian thực hiện là  $f(n)$  và  $f = O(g)$ . Ta nói thuật toán  $T$  có độ phức tạp  $g$ .

(hàm  $g$  chỉ là một chặn trên của  $f$ , vẫn có thể có cách ước lượng chặt hơn)

Định nghĩa: [cuuduongthancong.com](http://cuuduongthancong.com)

Ta nói  $f$  tương đương  $g$  nếu  $f = O(g)$  và  $g = O(f)$ , ta viết  $f \sim g$ .

Ví dụ: Thuật toán  $T$ , kích thước  $n$ , có thời gian chạy  $f(n) = \frac{1}{10}n^3 + 100n$

- Ta có thể chứng minh:  
 $f = O(n^3)$  và  $n^3 = O(f)$

$$\frac{1}{10} n^3 + 1000n \leq M n^3$$

$$\Rightarrow n^3 + 10000n \leq 10M n^3$$

$$\Rightarrow 10000n \leq (10M - 1)n^3$$

$$\Rightarrow 10000 \leq (10M - 1)n^2$$

Chọn  $M=1$ ,  $n_0=100 \Rightarrow f \sim n^3$

Ta nói “T có độ phức tạp tương đương  $n^3$ ”

- Một số tính chất: xét hai hàm  $f(n)$  và  $g(n)$

a) Nếu  $g(n) \neq 0$  khi  $n$  đủ lớn và  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$  tồn tại thì  $f=O(g)$ .

- Nếu  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq 0$  khi  $n$  đủ lớn thì  $f \sim g$

- Nếu  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$  thì  $f=O(g)$  nhưng  $g \neq O(f)$

Ví dụ:  $f(n) = (-1)^n n$  và  $g(n)=n+7$ ,  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$  không tồn tại do  $n$  chẵn hay lẻ.

Tuy nhiên  $\lim_{n \rightarrow \infty} \left| \frac{f(n)}{g(n)} \right| = 1 \Rightarrow f \sim g$

b) Nếu  $f$  là đa thức bậc  $\leq m$  thì  $f=O(n^m)$

c) Nếu  $f=O(g)$  và  $g=O(h)$  thì  $f=O(h)$

d) Một số công thức:

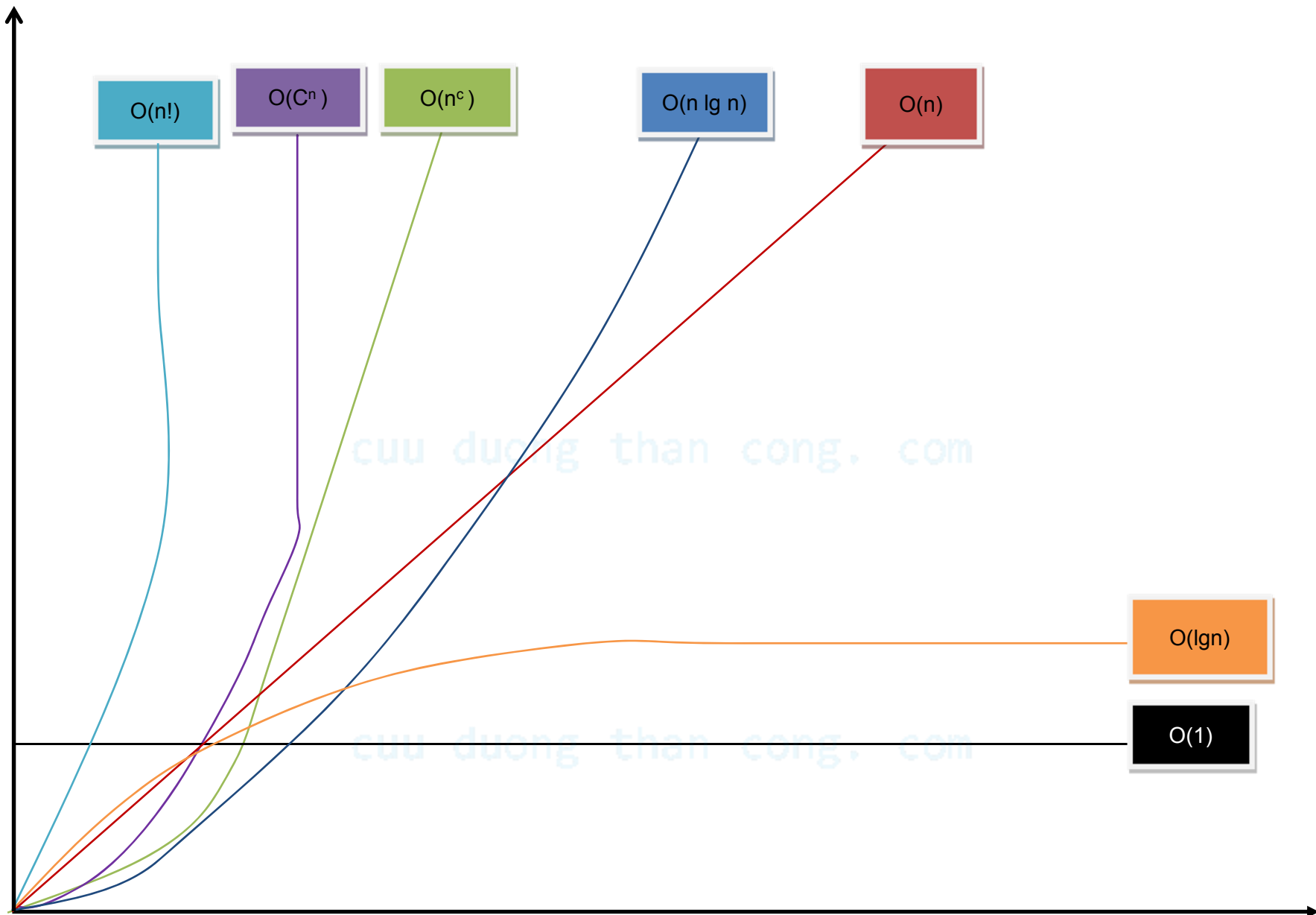
$$1 + \frac{1}{2} + \dots + \frac{1}{n} = \ln(n) + C + O\left(\frac{1}{n}\right)$$

với  $C=0.577216$  (hằng số Euler)

$$1^2 + \frac{1}{2^2} + \dots + \frac{1}{n^2} = ?$$

# Phân lớp các hàm

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





- **Ví dụ:** xét độ phức tạp khi xét một số nguyên dương  $n$  có phải là số nguyên tố hay không?
  - Kiểm tra các ước từ 2 đến  $n-1 \Rightarrow$  độ phức tạp là  $O(n)$
  - Nếu kiểm tra từ 2 đến  $\sqrt{n} \Rightarrow$  độ phức tạp là  $O(\sqrt{n})$
  - Nếu  $n$  khoảng vài tỷ và  $n=2^m$  với  $m$  là số bit lưu trữ, nếu chọn  $m$  là kích thước thuật toán thay cho  $n \Rightarrow$  độ phức tạp của thuật toán trên trong hai trường hợp là  $O(2^m)$  và  $O(2^{m/2})$  là hàm mũ.

# Vai trò của hằng số trong phân tích

- Thuật toán A và B có độ phức tạp

$$2n + \alpha \sqrt{n} + O\left(\frac{1}{n}\right)$$

$\alpha$  Và  $\beta$  có vai trò như thế nào?

$$\beta \sqrt{n} + O\left(\frac{1}{n}\right)$$

- Lý thuyết: do  $n$  khá lớn nên không đáng kể
- Thực nghiệm: đôi khi rất quan trọng  $\Rightarrow$  **cNh**  
**thận**

# Sự phụ thuộc/không phụ thuộc vào phân bố dữ liệu

- Xét bài toán A có thuật toán T có kích thước n
- Độ phức tạp của T:
  1. Hoàn toàn xác định theo n.  
Ví dụ: Tìm số lớn nhất của mảng các số nguyên.
  2. Ngẫu nhiên tùy theo phân bố của dữ liệu nhập.  
Ví dụ: Tìm phần tử x có hay không có trong tập dữ liệu.
- Cách giải quyết:
  1. Vận dụng các phép toán cơ bản để giải quyết.
  2. Ta phải xét :
    - a. Trường hợp xấu nhất (chậm nhất): chặn trên
    - b. Trường hợp tốt nhất (nhanh nhất): chặn dưới
    - c. Trung bình: vận dụng toán học (xác suất thống kê)Ví dụ: QuickSort

# Cách tính O

- **Quy tắc cộng:** Nếu  $K(n)$  và  $H(n)$  là thời gian thực hiện hai đoạn chương trình P và Q liên tiếp, với  $K(n)=O(f(n))$  và  $H(n)=O(g(n))$  thì thời gian thực hiện hai đoạn này là  $T(n)=O(\max(f(n),g(n)))$ .
- **Quy tắc nhân:** Nếu  $K(n)$  và  $H(n)$  là thời gian thực hiện hai đoạn chương trình P và Q lồng vào nhau, với  $K(n)=O(f(n))$  và  $H(n)=O(g(n))$  thì thời gian thực hiện hai đoạn này là  $T(n)=O(f(n).g(n))$ .