

# Nhập môn

## Phân tích độ phức tạp thuật toán

Chủ đề 3: Tiếp cận thực nghiệm để phân tích độ phức tạp thuật toán



KHOA CÔNG NGHỆ THÔNG TIN  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



## Tóm tắt nội dung chủ đề 3

Chủ đề này trình bày phương pháp tiếp cận thực nghiệm để khảo sát độ phức tạp của thuật toán.



## Nội dung chủ đề 3

- ☐ Yếu tố chi phối độ phức tạp tính toán
- ☐ Ý tưởng chung về đánh giá thực nghiệm
- ☐ Lưu ý trong thực nghiệm
- ☐ Thuật toán chỉ phụ thuộc kích thước dữ liệu
- ☐ Quy trình đánh giá thực nghiệm





# Yếu tố chi phối độ phức tạp tính toán

Độ phức tạp tính toán của mỗi thuật toán phụ thuộc vào hai yếu tố chính:

- (i) Kích thước của dữ liệu nhập
- (ii) Sự phân bố của dữ liệu nhập

Với thuật toán **T** giải bài toán **P**.

- ☐ Quy kích thước dữ liệu về một số tự nhiên  $n$  (thường cũng là kích thước bài toán).
- ☐ Hàm số  $g(T, n)$  phụ thuộc thuật toán  $T$ , kích thước  $n$  và sự phân bố của dữ liệu;

→ Hàm này đặc trưng cho độ phức tạp của  $T$ .

- ☐ Ta thường quan tâm: “*xấu nhất*”, “*tốt nhất*”, “*trung bình*” của thuật toán đang xét.



# Hoạt động của thuật toán

- ☐ *Trường hợp xấu nhất*: ứng với một bộ dữ liệu nhập nào đó mà  $g(T, n)$  có giá trị lớn nhất ;
- ☐ *Trường hợp tốt nhất*: ứng với một bộ dữ liệu nhập nào đó mà  $g(T, n)$  có giá trị nhỏ nhất ;
- ☐ *Trường hợp trung bình*: giá trị trung bình  $h(n)$  của  $g(T, n)$  khi chạy với tất cả các bộ dữ liệu có thể xảy ra (của bài toán đang xét) với mỗi kích thước  $n$  cố định.



## Ví dụ

- Bài toán sắp xếp mảng một chiều gồm  $n$  phần tử.
- Thuật toán Quicksort
  - Xấu nhất: mảng đã sắp xếp, số các thao tác tương đương  $n^2$ .
  - Tuy nhiên khả năng xảy ra trường hợp xấu nhất rất thấp (nói chung hiếm xảy ra)
  - Trung bình: khoảng  $n \cdot \log(n)$  thao tác
- Về mặt lý thuyết: độ phức tạp là  $O(n^2)$ , không thể hạ xuống, cũng không là  $\Theta(n^2)$ .



# Dạng thuật toán không phụ thuộc phân bố dữ liệu

- Một vài thuật toán có độ phức tạp tính toán
  - chỉ phụ thuộc vào kích thước dữ liệu nhập
  - không phụ thuộc vào sự phân bố của dữ liệu nhập

So sánh hai  
thuật toán này  
(Xem thêm trình  
bày trên bảng...)

```
max ← a[0]; i ← 1 ;
while i < n do
    if max < a[i] then
        max ← a[i] ; { α lần }
    endif
    i ← i+1 ;
endwhile;
```

```
max ← a[0]; i ← 1; count ← 0 ;
while i < n do
    if max < a[i] then
        max ← a[i] ;
    else
        count ← count+1 ;
    endif
    i ← i+1 ;
endwhile
```



## Ý tưởng chung về đánh giá thực nghiệm

- Chèn thêm một số lệnh vào cài đặt của thuật toán để đếm số thao tác chủ yếu (có vai trò quan trọng) của thuật toán
- Với mỗi kích thước  $n$  của dữ liệu nhập:
  - Chạy  $m$  lần chương trình đếm số thao tác, mỗi lần chạy:
    - Lấy ngẫu nhiên một bộ dữ liệu nhập kích thước  $n$
    - Ghi nhận, cộng dồn kết quả đếm từng thao tác trong mỗi lần chạy
  - Tính giá trị trung bình của từng thao tác cho  $m$  lần chạy
  - Ghi nhận lại số lượng trung bình của từng thao tác tương ứng với giá trị của kích thước  $n$ .
- Khảo sát với các giá trị tiêu biểu của  $n$ .





## Vài điểm quan trọng cần lưu ý

- Với mỗi kích thước  $n$  của bài toán, ta cần chọn một giá trị  $m$  thích hợp
  - Số  $n$  càng lớn thì  $m$  càng lớn ;
  - Lý tưởng thì  $m$  bằng số tổ hợp của tất cả những khả năng có thể có của những bộ dữ liệu kích thước  $n$  ;
  - Thực tế số tổ hợp quá lớn nên  $m$  phải chọn nhỏ hơn.
- Tính ngẫu nhiên của từng bộ dữ liệu thử nghiệm đóng vai trò rất quan trọng:
  - Khởi tạo bộ sinh ngẫu nhiên mỗi lần bắt đầu chạy cho một kích thước  $n$  (một lần ứng với mỗi giá trị  $n$ )
  - Lưu ý đến những điểm hạn chế của các hàm ngẫu nhiên trong thư viện để có điều chỉnh phù hợp



# Ví dụ về chèn thêm lệnh đếm

- Trong thuật toán (màu xanh) tìm phần tử lớn nhất ở phần trước:
  - Số lần thực hiện chỉ thị gán **max**  $\leftarrow$  **a[i]** là số  $\alpha$  thay đổi từ 0 đến n-1.
  - Những thao tác khác đều thực hiện n hay n-1 lần.

```
max  $\leftarrow$  a[0]; i  $\leftarrow$  1; alpha  $\leftarrow$  0 ;  
while i < n do  
    if max < a[i] then  
        max  $\leftarrow$  a[i]; alpha  $\leftarrow$  alpha + 1;  
    endif  
    i := i+1;  
endwhile
```

# Bình luận thêm thuật toán

- Thuật toán này có độ phức tạp là

$$f(n) = An + B + \alpha$$

Với A, B là hằng số không phụ thuộc  $n$ ; biến  $\alpha = 0, 1, \dots, n-1$ .

- Hiển nhiên  $f(n) = \Theta(n)$ , nên thuật toán có độ phức tạp tương đương tuyến tính.
- Tuy nhiên, biến  $\alpha$  lại thay đổi và có min, max, trung bình...

```
max ← a[0]; i ← 1 ;  
while i < n do  
    if max < a[i] then  
        max ← a[i] ; { α lần }  
    endif  
    i ← i+1 ;  
endwhile;
```

# Qui trình đánh giá thực nghiệm

Giả sử **Alg** là thuật toán cần được ước lượng.

**Bước 1** (làm thủ công).

Chèn thêm các lệnh vào **Alg** để có chương trình **Prg** đếm số thao tác tính toán thực hiện trong **Alg**.

**Bước 2.** Chọn một kích thước tối đa  $N\_MAX$  cho việc thí nghiệm và thực hiện vòng lặp sau đây:

Với mỗi kích thước  $n \in \{1, 2, \dots, N\_MAX\}$  thực hiện

Khởi động các biến tính trung bình cho mỗi loại phép tính về 0

Lặp lại quá trình sau  $M$  lần {với  $M$  khá lớn tùy theo  $n$ }

Chọn ngẫu nhiên một bộ dữ liệu nhập  $DATA(n)$  có kích thước  $n$

Chạy chương trình **Prg** cho bộ dữ liệu  $DATA(n)$  để đếm mỗi loại phép tính

Tăng tương ứng các biến tính trung bình theo kết quả vừa đếm

Cuối lặp

Chia các biến tính trung bình cho  $M$

Ghi nhận lại giá trị  $n$  và các biến tính trung bình

Cuối với mỗi



## Vài ghi chú

- Việc chọn các khoảng giá trị của  $n$  sẽ tùy vào thiết kế thực nghiệm
  - ▣ Có thể lấy liên tục các giá trị nhỏ
  - ▣ Lấy “nhảy cóc” khoảng nhỏ với giá trị vừa
  - ▣ Lấy bước nhảy khá lớn với giá trị lớn
- Ta thường dùng các công cụ kèm theo để vẽ đồ thị biến thiên của độ phức tạp  $f(n)$  theo  $n$  (tùy theo khoảng giá trị của  $n$ ).



# CÂU HỎI & THẢO LUẬN

