Phân tích và Thiết kế THUẬT TOÁN

Hà Đại Dương
duonghd@mta.edu.vn
Web: fit.mta.edu.vn/~duonghd

Bài 1 - Giới thiệu

PHÂN TÍCH VÀ THIẾT KẾ THUẬ TOÁN

NỘI DUNG

I. Giới thiệu
   1. Mục đích
   2. Nội dung môn học
   3. Hình thức Kiểm tra, Thi và đánh giá kết quả
   4. Tài liệu tham khảo

II. Thuật toán
   1. Định nghĩa
   2. Tính chất
   3. Biểu diễn

NỘI DUNG

III. Độ phức tạp thuật toán
   1. Giới tiệu
   2. Hướng tiếp cận
   3. Phân lớp độ phức tạp

IV. Bài tập

I. Giới thiệu

1. MỤC ĐÍCH

- Cung cấp kiến thức về việc đánh giá thuật toán
  - Lý thuyết
  - Thực nghiệm
- Kiến thức, kỹ thuật về giải quyết bài toán trên máy tính:
  - Trực tiếp
  - Gián tiếp
- Thiết kế thuật giải
  - Chia để trị
  - Quy hoạch động
  - Tìm kiếm cục bộ
  - …

I. Giới thiệu

2. NỘI DUNG

- Tổng quan về thuật toán và độ phức tạp của thuật toán
- Đánh giá thuật toán:
  - Lý thuyết (toán học sơ cấp)
  - Thực nghiệm
  - Đệ quy và phương pháp đánh giá
- Đánh giá một số thuật toán thông dụng
  - Thuật toán tìm kiếm
  - Thuật toán sắp xếp

I. Giới thiệu

2. NỘI DUNG

- Các phương pháp giải quyết bài toán trên máy tính
  - Trực tiếp
  - Gián tiếp
- Kỹ thuật thiết kế thuật toán
  - Chia để trị
  - Giải thuật tham lam
  - Quy hoạch động
  - Tìm kiếm cục bộ

I. Giới thiệu

3. HÌNH THỨC KIỂM TRA

- 10% Chuyên cần
- 20% Thường xuyên (bài tập, bài kiểm tra)
- 70% Thi cuối kỳ (vấn đáp): Sinh viên thực hiện 1 bài tập lớn với yêu cầu:
  - Cài đặt thuật toán cho bài toán đặt ra,
  - Chạy với dữ liệu phát sinh ngẫu nhiên, đếm số phép gán và so sánh, vẽ đồ thị, tính phương sai, độ lệch chuẩn -> Ước lượng độ phức tạp của thuật toán
  - Tính toán bằng lý thuyết và so sánh với thực nghiệm.
  - Viết báo cáo, vấn đáp trả lời các câu hỏi đặt ra.

I. Giới thiệu

4. TÀI LIỆU THAM KHẢO

- Slide bài giảng.
- Bài giảng Thiết kế và Đánh giá Thuật toán, Trần Xuân Sinh, NXB, ĐHQG, 2010.
- Cẩm nang thuật toán, Robert Sedgewich - Trần Đan Thư dịch (tái bản lần 2), NXB KHKT, 2006.
- Cấu trúc dữ liệu và giải thuật, Trần Xuân Lôi, NXB ĐH Quốc Gia, 2006.
- Giải một bài toán trên máy tính như thế nào (3 tập), Hoàng Kiếm, NXB Giáo dục, 2005.

I. Giới thiệu

4. TÀI LIỆU THAM KHẢO

- Giải thuật và lập trình (bài giảng chuyên đề), Lê Minh Hoàng, ĐHSP, 2002.
- Computer Algorithms Introduction to Design and Analysis, Addison-Wesley, 1988.
- Algorithms and Complexity, Herbert S. Wilf, University of Pennsylvania, Philadelphia 1999.
- Algorithm Design, Jon Kleinberg, Eva Tardos Pearson, 2006.

II. Thuật toán

1. ĐỊNH NGHĨA

Có nhiều cách phát biểu được chấp nhận, trong đó:

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

2) 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.

II. Thuật toán

2. TÍNH CHẤT

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

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

II. Thuật toán

2. TÍNH CHẤT

- 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?

II. Thuật toán

2. TÍNH CHẤT

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.

II. Thuật toán

2. TÍNH CHẤT

- Tính thực thi được, ví dụ:
  – Tính $\sqrt{-1}$?
- 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

II. Thuật toán

3. BIỂU DIỄN THUẬT TOÁN

– Ngôn ngữ tự nhiên
– Sơ đồ (lưu đồ) khối
– Mã giả (Pseudo-code)

II. Thuật toán

4. THUẬT GIẢI

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 thứ tự

III. Độ phức tạp thuật toán

1. GIỚI THIỆU

- Kích thước của bài toán **n** (có thể hiểu là số phần tử cần phải xử lý của bài toán)
  - Ví dụ: Sắp xếp một danh sách n sinh viên
  - Tìm phần tử X trong một mảng có n phần tử
  - …
- Với 1 bài toán có thể có nhiều thuật giải (chương trình)
  - Chương trình 1
  - Chương trình 2
- Chương trình (thuật giải) nào tốt?

III. Độ phức tạp thuật toán

1. GIỚI THIỆU

- Đánh giá độ phức tạp thời gian tính toán

(Vì thời gian còn phụ thuộc vào một máy tính cụ thể)

- Đánh giá theo tổng số các phép toán cơ bản (gán, so sánh)

- Vì việc đánh giá phụ thuộc vào **n** nên độ phức tạp thuật toán được hiểu như một hàm phụ thuộc vào **n, f(n)**

III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- Đánh giá **f(n)** như thế nào?

1. Hướng tiệm cận:
   - Lý thuyết
   - Thực nghiệm
2. Công cụ toán học:
   - Kỹ thuật sơ cấp
   - Hàm sinh
   - Hoán vị và nghịch thế

III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- 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

III. Độ phức tạp thuật toán

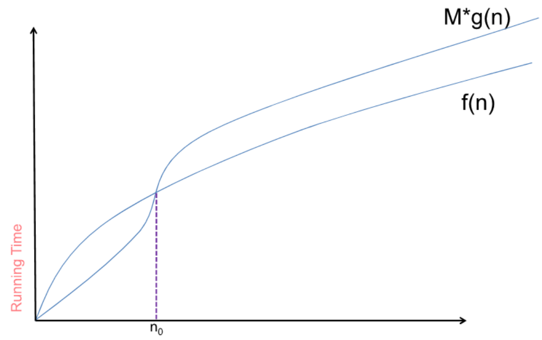
2. HƯỚNG TIẾP CẬN

- Ước lượng tiệm cận(lý thuyết):
  - Ý nghĩa: Phân lớp cấp độ lớn của các hàm f(n) khi n đủ lớn.
    Ký hiệu O (big O – O lớn)
  - Định nghĩa: Cho 2 hàm f,g : $N \rightarrow R$, ta nói $f = O(g)$ nếu $\exists n_0 \in N$ và $M>0$, sao cho $|f(n)| \leq M|g(n)|$, $\forall n \geq n_0$.

III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- Ước lượng tiệm cận(lý thuyết):



III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- Ước lượng tiệm cận(lý thuyết):
  - 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 $|f(n)| \leq 1.|g(n)|$, $\forall n \geq 1$.
    - 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 $|f(n)| \leq 1.|g(n)|$, $\forall n \geq 10000$

III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- Ước lượng tiệm cận(lý thuyết):
  - 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)

III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- Ước lượng tiệm cận(lý thuyết):
  - Thuật toán T có thời gian thực hiện là f(n) và f = O(g). Ta nói thuật tóan 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:

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

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

III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- Ước lượng tiệm cận(lý thuyết):
  - Ta có thể chứng minh:
    $f=O(n^3)$ và $n^3=O(f)$

$$\frac{1}{10}n^3 + 100n \le Mn^3$$
$$\Rightarrow \quad n^3 + 1000n \le 10Mn^3$$
$$\Rightarrow \quad 1000n \le (10M - 1)n^3$$
$$\Rightarrow \quad 1000 \le (10M - 1)n^2$$

Chọn M=1, n_0=100 ⇒ f ~ n^3

Ta nói "T có độ phức tạp tương đương n^3"

III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- Ước lượng tiệm cận(lý thuyết):
  - Một số tính chất: xét hai hàm f(n) và g(n)

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

    - Nếu $\lim_{n\to\infty}\frac{f(n)}{g(n)} \neq 0$ khi n đủ lớn thì f~g
    - Nếu $\lim_{n\to\infty}\frac{f(n)}{g(n)} = 0$ thì f=O(g) nhưng g≠O(f)

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

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

III. Độ phức tạp thuật toán

2. HƯỚNG TIẾP CẬN

- Ước lượng tiệm cận(lý thuyết):

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)

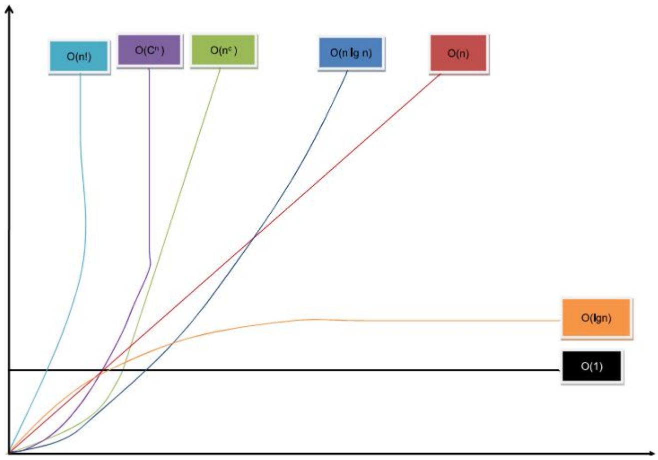
III. Độ phức tạp thuật toán

3. PHÂN LỚP CÁC HÀM

<table>
<tr><th>Dạng O</th><th>Tên Phân loại</th><th></th></tr>
<tr><td>O(1)</td><td>Hằng</td><td></td></tr>
<tr><td>O($\log_2$ (n))</td><td>logarit</td><td></td></tr>
<tr><td>O($\sqrt{n}$)</td><td rowspan="4">Căn thức</td><td></td></tr>
<tr><td>O($\sqrt[3]{n}$)</td><td></td></tr>
<tr><td>…</td><td></td></tr>
<tr><td>O($\sqrt[m]{n}$)</td><td></td></tr>
<tr><td>O(n)</td><td>Tuyến tính</td><td rowspan="5">Đa thức</td></tr>
<tr><td>O(n^2)</td><td>Bình phương</td></tr>
<tr><td>O(n^3)</td><td>Bậc ba</td></tr>
<tr><td>…</td><td></td></tr>
<tr><td>O(n^m)</td><td>Đa thức</td></tr>
<tr><td>O(c^n), với c>1</td><td>Mũ</td><td rowspan="2">Độ phức tạp lớn</td></tr>
<tr><td>O(n!)</td><td>Giai thừa</td></tr>
</table>

III. Độ phức tạp thuật toán

3. PHÂN LỚP CÁC HÀM



III. Độ phức tạp thuật toán

3. PHÂN LỚP CÁC HÀM

- 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.

III. Độ phức tạp thuật toán

4. MỘT SỐ VẤN ĐỀ KHÁC

- 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): chận trên
       b. Trường hợp tốt nhất (nhanh nhất): chận dưới
       c. Trung bình: vận dụng toán học (xác suất thống kê)
       Ví dụ: QuickSort

III. Độ phức tạp thuật toán

4. MỘT SỐ VẤN ĐỀ KHÁC

- Tính O (lớn) dựa trên quy tắc Cộng/Nhân
  - Quy tắc cộng: Nếu K(n) và H(n) là thời gian thực hiện hai đọan 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 đọan 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)).

NỘI DUNG BÀI HỌC

I. Giới thiệu
II. Thuật toán
III. Độ phức tạp thuật toán

IV. Bài tập

1. Nêu định nghĩa, tính chất và các cách thức biểu diễn thuật toán
2. Cho các bài toán sau:
   a) Tính nghiệm phương trình bậc 2: $ax^2+bx+c=0$, $a\neq 0$.
   b) Tính tổng bình phương của *n* số tự nhiên đầu tiên.
   c) Tìm số có giá trị *x* trong dãy $x_1, x_2,\ldots,x_n$.
   d) Tìm số có giá trị lớn nhất trong dãy $x_1, x_2,\ldots,x_n$.

   Hãy tìm thuật toán để giải bài toán trên, mô tả các thuật toán sử dụng ngôn ngữ tự nhiên và chỉ ra các tính chất của thuật toán đó.
3. Mô tả các thuật toán trong bài 2 dạng sơ đồ khối.
4. Mô tả các thuật toán trong bài 2 dạng giả mã.