

Design and Analysis of Algorithms

Lecture 1 Introduction

Lecturer: Tong Minh Duc
ductm@mta.edu.vn

1

Nội dung

1. Giới thiệu
2. Thuật toán
 - Khái niệm thuật toán
 - Biểu diễn thuật toán
 - Tính đúng đắn và hiệu quả của TT
3. Đánh giá độ phức tạp TT
 - Độ phức tạp của TT
 - Đánh giá bằng thực nghiệm

2/22/2023

2

Nội dung

- 1. Giới thiệu**
2. Thuật toán
3. Độ phức tạp thuật toán

2/22/2023

3

Mục đích

- Cung cấp kiến thức về việc đánh giá thuật toán
 - Lý thuyết
 - Thực nghiệm
- Thiết kế thuật giải
 - Chia để trị
 - Tham lam
 - Quy hoạch động
 - ...

2/22/2023

4

Nội dung môn học

- Tổng quan về thuật toán và độ phức tạp của thuật toán
- Đánh giá thuật toán
- Thiết kế thuật toán
- Phương pháp thiết kế thuật toán
 - Trực tiếp
 - Chia để trị
 - Tham lam ...

2/22/2023

5

Hình thức kiểm tra

- 10% Chuyên cần
- 30% Thường xuyên (bài tập, bài kiểm tra)
- 60% Thi cuối kỳ (vấn đáp)
 - Mô tả bài toán
 - Thiết kế thuật toán
 - Đánh giá
 - Cài đặt
 - Báo cáo

2/22/2023

6

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, Đỗ 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

2/22/2023

7

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

2/22/2023

8

Nội dung

1. Giới thiệu
- 2. Thuật toán**
3. Độ phức tạp thuật toán

2/22/2023

9

Khái niệm

- Một thuật toán là một bản liệt kê các chỉ dẫn, các quy tắc cần thực hiện theo từng bước xác định nhằm giải quyết một bài toán đã cho trong một khoảng thời gian hữu hạn.
- Ví dụ: Mô tả thuật toán giải quyết bài toán tìm phần tử lớn nhất trong dãy có n số cho trước.

2/22/2023

10

Ví dụ 1

1. Chỉ số phần tử lớn nhất tạm thời (LNTT) = chỉ số phần tử đầu tiên;
2. So sánh số tiếp theo với giá trị lớn nhất tạm thời, nếu lớn hơn giá trị lớn nhất tạm thời thì đặt:
 - Chỉ số phần tử LNTT = chỉ số phần tử đó;
3. Lặp lại bước 2) nếu còn phần tử trong dãy.
4. Phần tử lớn nhất tạm thời ở thời điểm này chính là phần tử lớn nhất trong dãy.

2/22/2023

11

Ví dụ 2

- Mô tả dưới dạng giả mã

```

input:       $n$  và  $a(1), \dots, a(n)$ ;
output:      $i_0$ , chỉ số phần tử lớn nhất trong  $a$ ;
1)  $i_0 = 1$ ;
2) for ( $i = 2$ ;  $i \leq n$ ;  $i++$ )
           if ( $a(i_0) < a(i)$ )     $i_0 = i$ ;
3) output  $i_0$ ;
  
```

2/22/2023

12

Tính chất của TT ...

- 1. Tính chính xác:** để đảm bảo kết quả tính toán hay các thao tác mà máy tính thực hiện được là chính xác.
- 2. Tính rõ ràng:** Thuật toán phải được thể hiện bằng các câu lệnh minh bạch; các câu lệnh được sắp xếp theo thứ tự nhất định.

2/22/2023

13

Tính chất của TT ...

- 3. Tính khách quan:** Một thuật toán dù được viết bởi nhiều người trên nhiều máy tính vẫn phải cho kết quả như nhau.
- 4. Tính phổ dụng:** Thuật toán không chỉ áp dụng cho một bài toán nhất định mà có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.
- 5. Tính kết thúc:** Thuật toán phải gồm một số hữu hạn các bước tính toán.

2/22/2023

14

Biểu diễn thuật toán

- Có 3 cách biểu diễn thuật toán:
 - Dùng ngôn ngữ tự nhiên
 - Sơ đồ khối và
 - Giả mã.
- Dùng ngôn ngữ tự nhiên: mô tả các bước xử lý bằng ngôn ngữ viết.

2/22/2023

15

Mô tả dữ liệu vào/ra

- *Dữ liệu đầu vào*: Một thuật toán phải mô tả rõ các giá trị đầu vào từ một tập hợp các dữ liệu xác định. Ví dụ, dãy số nguyên $a(1), a(2), \dots, a(n)$, với $n < \infty$.
- *Dữ liệu đầu ra*: Từ một tập các giá trị đầu vào, thuật toán sẽ tạo ra các giá trị đầu ra. Các giá trị đầu ra chính là nghiệm của bài toán. Ví dụ, i_0 là chỉ số phần tử lớn nhất trong $a(1), \dots, a(n)$.

2/22/2023

16

Ví dụ

- Thuật toán tìm số lớn nhất

- 1) Dữ liệu vào: dãy có n phần tử
- 2) Dữ liệu ra: Chỉ số phần tử lớn nhất
- 3) Chỉ số phần tử lớn nhất tạm thời = chỉ số phần tử đầu tiên;
- 4) So sánh số tiếp theo với giá trị lớn nhất tạm thời GTLNTT, nếu lớn hơn GTLNTT thì đặt
Chỉ số phần tử lớn nhất tạm thời = chỉ số phần tử đó;
- 5) Lặp lại bước 2) nếu còn phần tử trong dãy.
- 6) Phần tử lớn nhất tạm thời ở thời điểm này chính là phần tử lớn nhất trong dãy.

2/22/2023

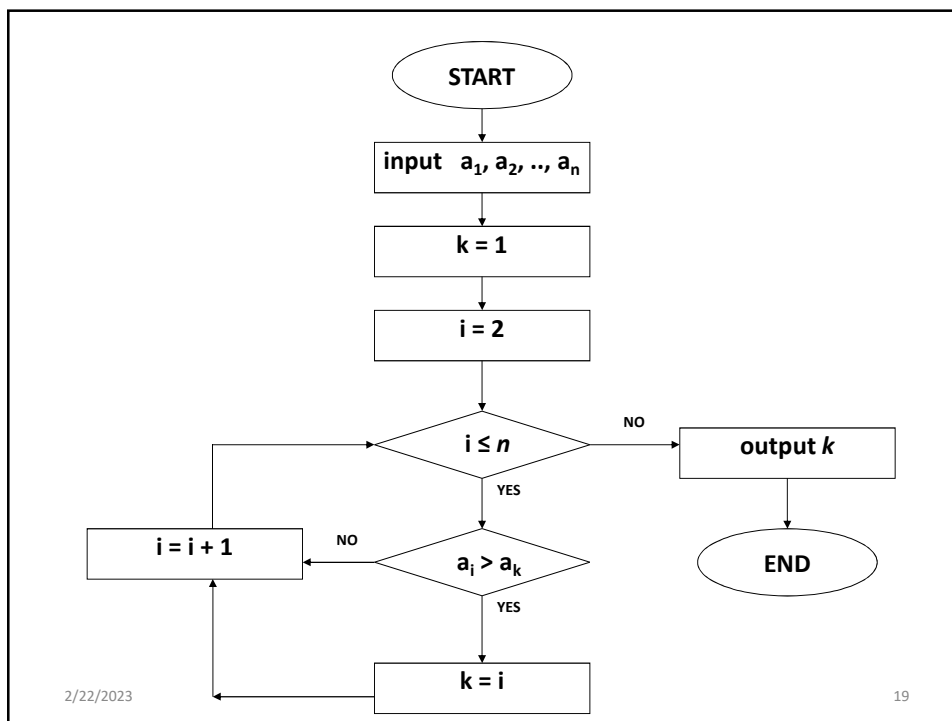
17

Sơ đồ khối

- Sử dụng bộ kí hiệu các khối để thể hiện thuật toán.
- Bộ kí hiệu:
 - Hộp chữ nhật: Các toán tử gán và phép toán tính toán;
 - Hình thoi: Phép toán so sánh cho kết quả thuộc tập: {đúng, sai}.
 - Đường kẻ + mũi tên: Chỉ thao tác tiếp theo;
 - Hình elip: Biểu thị sự bắt đầu hoặc kết thúc thuật toán

2/22/2023

18



Giải mã

- Sử dụng ngôn ngữ tự nhiên kết hợp với một ngôn ngữ lập trình.
- Cần tuân thủ quy tắc của một ngôn ngữ:
 - Có các cấu trúc cơ bản: tuần tự, lặp và rẽ nhánh.
 - Có hệ thống từ khóa, từ điển (phụ thuộc vào bài toán).
- Dễ hiểu, dễ cài đặt.

Ví dụ 1: Tìm phần tử lớn nhất ...

```

input:       $n$  và  $a(1), \dots, a(n)$ ;
output:      $i_0$ , chỉ số phần tử lớn nhất trong  $a$ ;
1)  $i_0 = 1$ ;
2) for ( $i = 2; i \leq n; i += 1$ )
    if ( $a(i_0) < a(i)$ )     $i_0 = i$ ;
3) output  $i_0$ ;
  
```

2/22/2023

21

Ví dụ 2: Tìm phần tử có giá trị b

- Tìm trong dãy có n số cho trước số có giá trị bằng b và in ra chỉ số i_0 của phần tử đầu tiên có giá trị bằng **b** . Nếu không tìm thấy thì chỉ số $i_0 = -1$

2/22/2023

22

Ví dụ 2: Tìm phần tử có giá trị b

- Tìm trong dãy có n số cho trước.

input: a_1, a_2, \dots, a_n, b ;

output: chỉ số i_0 phần tử đầu tiên có giá trị bằng b ,
nếu không có, $i_0 = -1$;

- 1) $i = 1$;
- 2) while $(i \leq n) \ \& \ (a_i \neq b) \quad i++$;
- 3) if $(i \leq n) \ i_0 = i$;
 else $i_0 = -1$;
- 4) output i_0 ;

2/22/2023

23

Chất lượng biểu diễn thuật toán

1. Đúng với ý tưởng đặt ra của bài toán
2. Đơn giản, dễ hiểu
3. Dễ cài đặt

2/22/2023

24

Tính đúng đắn của thuật toán

- PP lý thuyết, PP thực nghiệm
- Phương pháp lý thuyết:
 - Chứng minh thuật toán cho ra kết quả phù hợp với bài toán
 - Thuật toán kết thúc và cho ra kết quả
 - Kết quả phù hợp với yêu cầu của bài toán
 - Ví dụ:
 - Các thuật toán dựa trên qui hoặc động
 - Các thuật toán vét cạn ...

2/22/2023

25

Tính đúng đắn của thuật toán

- Phương pháp thực nghiệm:
 - Thuật toán có thể được xây dựng trên một ý tưởng dạng trực giác, là gần đúng hoặc không chứng minh được tính đúng đắn của thuật toán bằng phương pháp lý thuyết.
 - Thuật toán được chương trình hóa và được thực hiện trên một tập dữ liệu đủ lớn, bao hàm các trường hợp có thể xảy ra đối với bài toán ban đầu.

2/22/2023

26

Tính hiệu quả

1. Thời gian: Chi phí cho thời gian tính toán ít hơn so với các thuật toán giải quyết cùng bài toán.
2. Bộ nhớ: Chiếm dụng bộ nhớ ít hơn so với các thuật toán giải quyết cùng bài toán.
3. Độ chính xác: Nếu là cung cấp lời giải gần đúng thì gần với lời giải đúng hơn so với thuật toán giải quyết cùng bài toán

2/22/2023

27

Tính hiệu quả

- Tùy theo bài toán, mục đích mà xét các tiêu chí nói trên.
- Tùy theo chất lượng của thuật toán mà có thể xét trên mọi tập dữ liệu thử nghiệm, hoặc trên một số tập dữ liệu minh họa cho một vài khía cạnh nào đó.

2/22/2023

28

Nội dung

1. Giới thiệu
2. Thuật toán
3. **Độ phức tạp thuật toán**

2/22/2023

29

Độ tăng của hàm ...

- Cho f và g , $f: \mathbb{R} \rightarrow \mathbb{R}$, $g: \mathbb{R} \rightarrow \mathbb{R}$.
- **Định nghĩa:** $f(x)$ là O-lớn của $g(x)$ khi $x \rightarrow \infty$, kí hiệu $f(x) = O(g(x))$, nếu $\exists C > 0$ và $N > 0$ sao cho $\forall x > N$ thì $|f(x)| \leq C \cdot |g(x)|$.
- Ví dụ:
 - $f(x) = x^2 + 2x + 3 = O(x^2)$, vì với mọi $x > 1$ ta có $f(x) \leq x^2 + 2x^2 + 3x^2 = 6x^2$. Ngược lại, $x^2 = O(f(x))$ vì hiển nhiên là với mọi $x > 0$ ta có $x^2 < f(x)$.

2/22/2023

30

Một số tính chất của O-lớn

- $f(x) = a_0 + a_1x^1 + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n = O(x^n)$
- Chứng minh

Kí hiệu $C = |a_0| + |a_1| + |a_2| + \dots + |a_{n-1}| + |a_n|$.

Với $x > 1$ ta có $x^k < x^n$, với $k < n$, suy ra

$$\begin{aligned} |f(x)| &< |a_0 + a_1x^1 + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n| &&\leq \\ &|a_0| + |a_1x^1| + |a_2x^2| + \dots + |a_{n-1}x^{n-1}| + |a_nx^n| &&\leq \\ &|a_0| + |a_1| \cdot x + |a_2| \cdot x^2 + \dots + |a_{n-1}| \cdot x^{n-1} + |a_n| \cdot x^n &&\leq \\ &(|a_0| + |a_1| + |a_2| + \dots + |a_{n-1}| + |a_n|) \cdot x^n &&\leq C \cdot x^n. \text{ (đpcm)} \end{aligned}$$

2/22/2023

31

Một số tính chất của O-lớn

- Ở đây ta dùng một đại lượng tổng quát là tài nguyên cần dùng $R(n)$.
- Đó có thể là số lượng phép tính (có thể tính cả số lần truy nhập bộ nhớ, hoặc ghi vào bộ nhớ);
- Cũng có thể là thời gian thực hiện chương trình (độ phức tạp về thời gian)
- Hoặc dung lượng bộ nhớ cần phải cấp để chạy chương trình (độ phức tạp về không gian).

2/22/2023

32

Quy tắc bỏ hằng số

- $T(n) = O(c \cdot f(n)) = O(f(n))$ với c là một hằng số dương

2/22/2023

33

Quy tắc lấy max

- $T(n) = O(f(n) + g(n)) = O(\max(f(n), g(n)))$

2/22/2023

34

Quy tắc cộng

- $T1(n) = O(f(n))$ $T2(n) = O(g(n))$
- $T1(n) + T2(n) = O(f(n) + g(n))$

2/22/2023

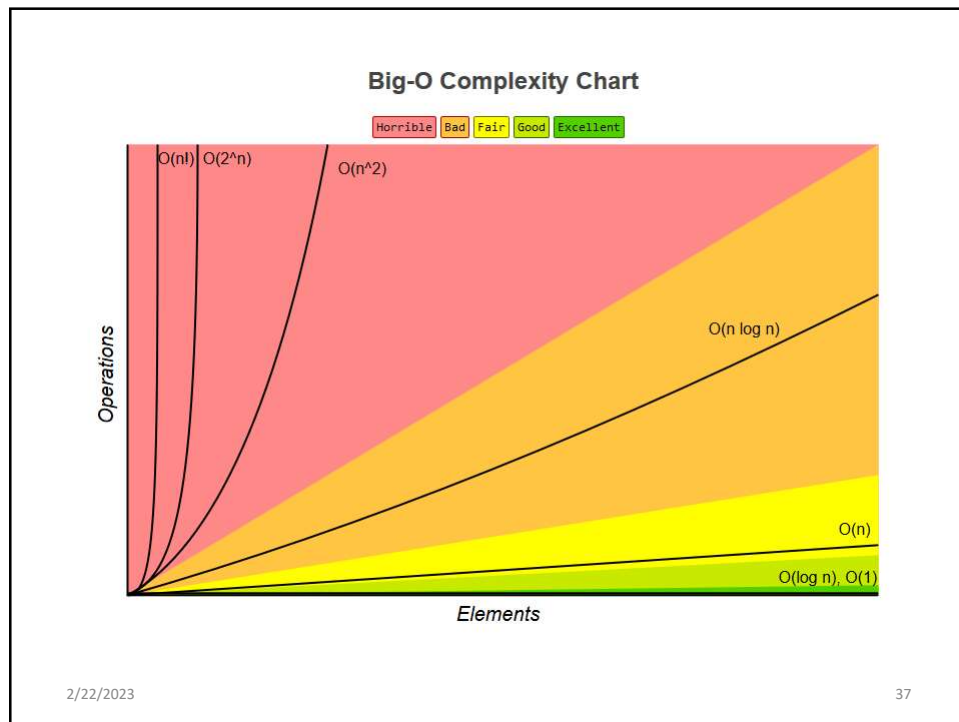
35

Quy tắc nhân

- Đoạn chương trình có thời gian thực hiện $T(n)=O(f(n))$
- Nếu thực hiện $k(n)$ lần đoạn chương trình với $k(n) = O(g(n))$ thì độ phức tạp sẽ là $O(g(n).f(n))$

2/22/2023

36



Ví dụ 1

- $s = n * (n - 1) / 2$;
- Trong ví dụ trên, độ phức tạp của thuật toán là $O(1)$

Ví dụ 2

```

s = 0;           // O(1)
for (i=0; i<=n;i++){
    p = 1;       // O(1)
    for (j=1;j<=i;j++)
        p = p * j;    // O(1)
    s = s+p;       // O(1)
}

```

- Số lần thực hiện phép toán $p = p * j$ là $n(n-1) / 2$
- Độ phức tạp của đoạn code này là $O(1) + O(n) + O(n(n-1)/2) + O(n)$
- $= O(n^2)$

2/22/2023

39

Ví dụ 3

```

for (i= 1;i<=n;i++) {
    for (u= 1;u<=m;u++)
        for (v= 1;v<=n;v++)
            //lệnh
    for j:= 1 to x do
        for k:= 1 to z do
            //lệnh
}

```

Độ phức tạp của thuật toán này là:
 $O(n * \max(n * m, x * z))$

2/22/2023

40

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 lớn thứ nhất và thứ 2 của dãy có các phần tử khác nhau với độ phức tạp thấp nhất có thể
 - d) Tìm số có giá trị x trong dãy x_1, x_2, \dots, x_n .
 - e) Tìm số có giá trị lớn nhất trong dãy x_1, x_2, \dots, 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.

2/22/2023

41

Bài tập

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

2/22/2023

42