

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NÔNG NGHIỆP HÀ NỘI

VŨ KIM THÀNH

TOÁN RỜI RẠC

(Giáo trình dành cho sinh viên ngành công nghệ thông tin)

Hà nội 2008

MỤC LỤC

Lời nói đầu	5
Chương 1. THUẬT TOÁN	7
1. Định nghĩa	7
2. Mô tả thuật toán bằng lưu đồ	8
3. Mô tả thuật toán bằng ngôn ngữ phòng Pascal	9
4. Độ phức tạp của thuật toán	14
5. Thuật toán tìm kiếm	18
6. Thuật toán đệ quy	19
7. Một số thuật toán về số nguyên	23
BÀI TẬP CHƯƠNG 1	28
Chương 2. BÀI TOÁN ĐẾM	32
1. Nguyên lý cộng và nguyên lý nhân	32
2. Chỉnh hợp. Hoán vị. Tổ hợp.	35
3. Nguyên lý bù trừ	42
4. Giải các hệ thức truy hồi	44
5. Bài toán liệt kê.	51
6. Bài toán tồn tại	61
BÀI TẬP CHƯƠNG 2	64
Chương 3. CÁC KHÁI NIỆM CƠ BẢN VỀ ĐỒ THỊ	69
1. Các định nghĩa về đồ thị và biểu diễn hình học của đồ thị	69
2. Biểu diễn đồ thị bằng đại số	79
3. Sự đẳng cấu của các đồ thị	82
4. Tính liên thông trong đồ thị	84
5. Số ổn định trong, số ổn định ngoài và nhân của đồ thị	88
6. Sắc số của đồ thị	91
BÀI TẬP CHƯƠNG 3	93
Chương 4. ĐỒ THỊ EULER, ĐỒ THỊ HAMILTON, ĐỒ THỊ PHẪNG	98
1. Đồ thị Euler	98
2. Đồ thị Hamilton	103
3. Đồ thị phẳng	108
BÀI TẬP CHƯƠNG 4	113
Chương 5. CÂY VÀ MỘT SỐ ỨNG DỤNG CỦA CÂY	117
1. Cây và các tính chất cơ bản của cây	118
2. Cây nhị phân và phép duyệt cây	122
3. Một vài ứng dụng của cây	126

4. Cây khung (cây bao trùm) của đồ thị	131
5. Hệ chu trình độc lập	134
6. Cây khung nhỏ nhất	136
BÀI TẬP CHƯƠNG 5	142
Chương 6. MỘT SỐ BÀI TOÁN TỐI ƯU TRÊN ĐỒ THỊ	147
1. Bài toán đường đi ngắn nhất trong đồ thị	147
2. Tâm, Bán kính, Đường kính của đồ thị	152
3. Mạng và Luồng	153
4. Bài toán du lịch	160
BÀI TẬP CHƯƠNG 6	166
Chương 7. ĐẠI SỐ BOOLE	172
1. Hàm Boole	172
2. Biểu thức Boole	174
3. Định nghĩa đại số Boole theo tiên đề	176
4. Biểu diễn các hàm Boole	177
5. Các cổng logic	183
6. Tối thiểu hoá hàm Boole	185
BÀI TẬP CHƯƠNG 7	193
Phụ chương. ĐẠI CƯƠNG VỀ TOÁN LOGIC	197
1. Logic mệnh đề	197
2. Công thức đồng nhất đúng và công thức đồng nhất bằng nhau trong logic mệnh đề	201
3. Điều kiện đồng nhất đúng trong logic mệnh đề	205
4. Logic vị từ	208
BÀI TẬP PHỤ CHƯƠNG	213
Một số bài tập làm trên máy tính	216
Một số thuật ngữ dùng trong giáo trình	218
Tài liệu tham khảo	221

LỜI NÓI ĐẦU

Toán Rời rạc (Discrete mathematics) là môn toán học nghiên cứu các đối tượng rời rạc. Nó được ứng dụng trong nhiều ngành khoa học khác nhau, đặc biệt là trong tin học bởi quá trình xử lý thông tin trên máy tính thực chất là một quá trình rời rạc.

Phạm vi nghiên cứu của Toán Rời rạc rất rộng, có thể chia thành các môn học khác nhau. Theo quy định của chương trình môn học, giáo trình này đề cập đến các lĩnh vực: Thuật toán và bài toán đếm; Lý thuyết đồ thị; Đại số Logic và được chia thành 8 chương:

- Chương 1 đề cập đến một trong các vấn đề cơ bản nhất của *Thuật toán* đó là độ phức tạp về thời gian của thuật toán.
- Chương 2 nói về các nguyên lý cơ bản của *Bài toán đếm*.
- Các chương 3, 4, 5 và 6 trình bày về *Lý thuyết đồ thị và các ứng dụng*. Đây là phần chiếm tỷ trọng nhiều nhất của giáo trình. Trong đó có các chương về các khái niệm cơ bản của đồ thị, các đồ thị đặc biệt như đồ thị Euler, đồ thị Hamilton, đồ thị phẳng, Cây cùng các ứng dụng của các đồ thị đặc biệt này. Riêng chương 6 dành cho một vấn đề trọng là một số bài toán tối ưu trên đồ thị hoặc bài toán tối ưu được giải bằng cách ứng dụng lý thuyết đồ thị.
- Chương 7 là các kiến thức cơ bản về *Đại số Boole*, một công cụ hữu hiệu trong việc thiết kế các mạch điện, điện tử.

Cuối giáo trình là phụ chương: *Những khái niệm cơ bản về toán Logic* để người học có thể tự nghiên cứu thêm về Toán Logic.

Trong mỗi chương chúng tôi cố gắng trình bày các kiến thức cơ bản nhất của chương đó cùng các thí dụ minh họa cụ thể. Vì khuôn khổ số tiết học nên chúng tôi lược bỏ một số chứng minh phức tạp. Cuối mỗi chương đều có các bài tập để người học ứng dụng, kiểm chứng các lý thuyết đã học, đồng thời cũng cung cấp một số đáp số của các bài tập đã cho.

Cũng cần nói thêm rằng toán Rời rạc không chỉ được ứng dụng trong tin học mà còn được ứng dụng trong nhiều ngành khoa học khác. Bởi vậy giáo trình cũng có ích cho những ai cần quan tâm đến các ứng dụng khác của môn học này.

Tác giả xin chân thành cảm ơn các bạn đồng nghiệp đã động viên và góp ý cho việc biên soạn giáo trình này. Đặc biệt chúng tôi xin cảm ơn Nhà giáo ưu tú Nguyễn Đình Hiền đã hiệu đính và cho nhiều ý kiến đóng góp bổ ích và thiết thực.

Vì trình độ có hạn và giáo trình được biên soạn lần đầu nên không tránh khỏi các thiếu sót. Tác giả rất mong nhận được các ý kiến đóng góp của các đồng nghiệp và bạn đọc về các khiếm khuyết của cuốn sách.

TÁC GIẢ

CHƯƠNG 1.

THUẬT TOÁN

1. Định nghĩa.
2. Mô tả thuật toán bằng lưu đồ.
3. Mô tả thuật toán bằng ngôn ngữ phỏng Pascal.
 - 3.1. Câu lệnh Procedure (thủ tục) hoặc Function (hàm).
 - 3.2. Câu lệnh gán.
 - 3.3. Khối câu lệnh tuần tự.
 - 3.4. Câu lệnh điều kiện.
 - 3.5. Các câu lệnh lặp.
4. Độ phức tạp của thuật toán.
 - 4.1. Khái niệm độ tăng của hàm.
 - 4.2. Độ tăng của tổ hợp các hàm.
 - 4.3. Độ phức tạp của thuật toán.
5. Thuật toán tìm kiếm
 - 5.1. Thuật toán tìm kiếm tuyến tính (còn gọi là thuật toán tìm kiếm tuần tự).
 - 5.2. Thuật toán tìm kiếm nhị phân.
6. Thuật toán đệ quy.
 - 6.1. Công thức truy hồi.
 - 6.2. Thuật toán đệ quy.
 - 6.3. Đệ quy và lặp
7. Một số thuật toán về số nguyên.
 - 7.1. Biểu diễn các số nguyên.
 - 7.2. Cộng và nhân trong hệ nhị phân.

1. Định nghĩa

Thuật toán (algorithm) là một dãy các quy tắc nhằm xác định một dãy các thao tác trên các đối tượng sao cho **sau một số hữu hạn** bước thực hiện sẽ **đạt được mục tiêu đặt ra**.

Từ định nghĩa của thuật toán cho thấy các đặc trưng (tính chất) cơ bản của thuật toán là:

a. *Yếu tố vào, ra:*

- Đầu vào (Input): Mỗi thuật toán có một giá trị hoặc một bộ giá trị đầu vào từ một tập xác định đã được chỉ rõ.
- Đầu ra (Output): Từ các giá trị đầu vào, thuật toán cho ra các giá trị cần tìm gọi là kết quả của bài toán.

b. Tính dừng:

Sau một số hữu hạn bước thao tác thuật toán phải kết thúc và cho kết quả .

c. Tính xác định:

Các thao tác phải rõ ràng, cho cùng một kết quả dù được xử lý trên các bộ xử lý khác nhau (hai bộ xử lý trong cùng một điều kiện không thể cho hai kết quả khác nhau).

d. Tính hiệu quả

Sau khi đưa dữ liệu vào cho thuật toán hoạt động phải đưa ra kết quả như ý muốn.

e. Tính tổng quát

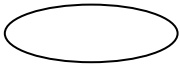
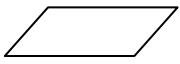
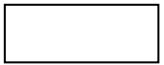
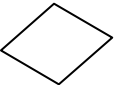
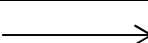
Thuật toán phải được áp dụng cho mọi bài toán cùng dạng chứ không phải chỉ cho một tập đặc biệt các giá trị đầu vào.

Có nhiều cách mô tả thuật toán như: Dùng ngôn ngữ tự nhiên; dùng lưu đồ (sơ đồ khối); dùng một ngôn ngữ lập trình nào đó (trong giáo trình này dùng loại ngôn ngữ mô tả gần như ngôn ngữ lập trình Pascal và gọi là phỏng Pascal); ...

2. Mô tả thuật toán bằng lưu đồ

Sau khi có thuật toán để giải bài toán, trước khi chuyển sang ngôn ngữ lập trình, người ta thường phải thể hiện thuật toán dưới dạng sơ đồ. Sơ đồ đó gọi là lưu đồ của thuật toán. Các ký hiệu quy ước dùng trong lưu đồ được trình bày trong bảng 1.

Bảng 1. Các ký hiệu quy ước dùng trong lưu đồ thuật toán

Tên ký hiệu	Ký hiệu	Vai trò của ký hiệu
Khối mở đầu hoặc kết thúc		Dùng để mở đầu hoặc kết thúc thuật toán
Khối vào ra		Đưa dữ liệu vào và in kết quả
Khối tính toán		Biểu diễn các công thức tính toán và thay đổi giá trị các đối tượng
Khối điều kiện		Kiểm tra các điều kiện phân nhánh
Chương trình con		Gọi các chương trình con
Hướng đi của thuật toán		Hướng chuyển thông tin, liên hệ giữa các khối

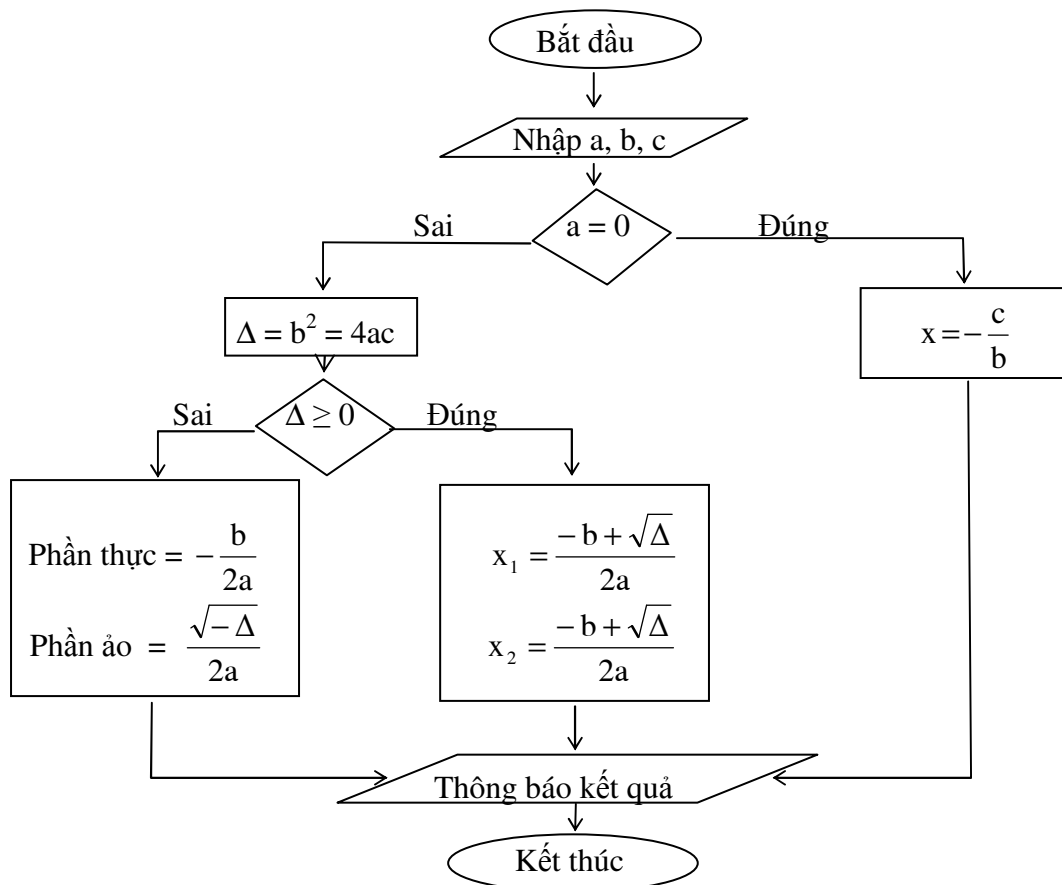
Thí dụ: Thuật toán giải phương trình bậc hai $ax^2 + bx + c = 0$ gồm các bước sau:

- 1) Xác định các hệ số a, b, c (thông tin đầu vào)
- 2) Kiểm tra hệ số a:

- Nếu a = 0: Phương trình đã cho là phương trình bậc nhất, nghiệm là: $x = -\frac{c}{b}$.
- Nếu a ≠ 0: Chuyển sang bước 3.

- 3) Tính biệt thức $\Delta = b^2 - 4ac$.
- 4) Kiểm tra dấu của biệt thức Δ
 - Nếu $\Delta \geq 0$: Phương trình có nghiệm thực
 - Nếu $\Delta < 0$: Phương trình có nghiệm phức
- 5) In kết quả

Lưu đồ của thuật toán được trình bày trong hình 1



Hình 1. Lưu đồ giải phương trình bậc hai

3. Mô tả thuật toán bằng ngôn ngữ phỏng Pascal

Để giải bài toán trên máy tính điện tử phải viết chương trình theo một ngôn ngữ lập trình nào đó (Pascal, C, Basic, ...). Mỗi ngôn ngữ lập trình có một quy tắc cấu trúc riêng. Để thay việc mô tả thuật toán bằng lời, có thể mô tả thuật toán bằng các cấu trúc lệnh tương tự như ngôn ngữ lập trình Pascal và gọi là ngôn ngữ phỏng Pascal.

Các câu lệnh chính dùng để mô tả thuật toán gồm có: Procedure hoặc Function; câu lệnh gán; các câu lệnh điều kiện; các vòng lặp. Ngoài ra khi cần giải thích các câu lệnh bằng lời, có thể để các lời giải thích trong dấu (* ... *) hoặc {...}.

Nghĩa là ngôn ngữ phỏng Pascal hoàn toàn tương tự ngôn ngữ lập trình Pascal, nhưng không có phần khai báo. Tuy nhiên, phải nêu rõ đầu vào (Input) và đầu ra (output) của thuật toán.

3.1. Câu lệnh Procedure (thủ tục) hoặc Function (hàm)

Đứng ngay sau câu lệnh này là tên của thủ tục hoặc tên hàm. Các bước thực hiện của thuật toán được mô tả trong thủ tục (hàm) được bắt đầu bởi từ khóa begin và kết thúc bởi từ khóa end.

Thí dụ 1

```
Function Max(a, b, c) (* Hàm tìm số lớn nhất trong 3 số a, b, c *)  
Begin  
  (* thân hàm *)  
End;
```

Thí dụ 2

```
Procedure Giai_phuong_trinh_bac_hai (* Thủ tục giải phương trình bậc hai *)  
Begin  
  (* thân thủ tục *)  
End;
```

Chú ý rằng, khi mô tả thuật toán bằng function, trước khi kết thúc (end) thuật toán phải trả về (ghi nhận) giá trị của function đó.

3.2. Câu lệnh gán

Dùng để gán giá trị cho các biến. Cách viết:
Tên biến := giá trị gán

Thí dụ: x := a; (*biến x được gán giá trị a*)
max := b; (*biến max được gán giá trị b*)

3.3. Khối câu lệnh tuần tự

Được mở đầu bằng từ khóa **begin** và kết thúc bằng **end** như sau:

```
begin  
  Câu lệnh 1;  
  Câu lệnh 2;  
  ... ..  
  Câu lệnh n;  
end;
```

Các lệnh được thực hiện tuần tự từ câu lệnh thứ nhất đến câu lệnh cuối cùng.

3.4. Câu lệnh điều kiện

Có hai dạng: dạng đơn giản và dạng lựa chọn.

a. Dạng đơn giản: Cách viết:

if <điều kiện> **then** câu lệnh hoặc khối câu lệnh;

Khi thực hiện, điều kiện được kiểm tra, nếu điều kiện thỏa mãn thì câu lệnh (khối câu lệnh) được thực hiện, nếu điều kiện không thỏa mãn thì lệnh bị bỏ qua.

b. Dạng lựa chọn: Cách viết:

if <điều kiện> **then** câu lệnh hoặc khối câu lệnh 1 **else** câu lệnh hoặc khối câu lệnh 2;

Khi thực hiện, điều kiện được kiểm tra, nếu điều kiện thỏa mãn thì câu lệnh (khối câu lệnh) 1 được thực hiện, nếu điều kiện không được thỏa mãn thì câu lệnh (khối câu lệnh) 2 được thực hiện.

Thí dụ 1. Thuật toán tìm số lớn nhất trong 3 số thực a, b, c.

- Đầu tiên cho max = a;
- So sánh max với b, nếu $b > \text{max}$ thì max = b;
- So sánh max với c, nếu $c > \text{max}$ thì max = c.

Function max(a,b,c)

Input: 3 số thực a,b,c;

Output: Số lớn nhất trong 3 số đã nhập;

Begin

 x := a;

 if $b > x$ then x:= b;

 if $c > x$ then x:= c;

 max := x;

End;

Thí dụ 2. Thuật toán giải phương trình bậc hai $ax^2 + bx + c = 0$

Procedure Giai_phuong_trinh_bac2;

Input: Các hệ số a, b, c;

Output: Nghiệm của phương trình;

begin

 if $a = 0$ then x := -c/b;

 if $a \neq 0$ then

 begin

$\Delta := b^2 - 4ac$;

 if $\Delta \geq 0$ then

 begin

$x_1 = (-b - \sqrt{\Delta})/2a$;

$x_2 = (-b + \sqrt{\Delta})/2a$;

 end

 else

 begin

 Phần thực := -b/2a;

 Phần ảo := $(\sqrt{-\Delta})/2a$;

 end;

 end;

end;

3.5. Các câu lệnh lặp

Có hai loại: Loại có bước lặp xác định và loại có bước lặp không xác định.

a. Loại có bước lặp xác định: Cách viết như sau:

for biến điều khiển := giá trị đầu **to** giá trị cuối **do** câu lệnh hoặc khối câu lệnh;

Khi thực hiện, biến điều khiển được kiểm tra, nếu biến điều khiển nhỏ hơn hoặc bằng giá trị cuối thì câu lệnh (khối câu lệnh) được thực hiện. Tiếp đó biến điều khiển sẽ tăng thêm 1 đơn vị và quá trình được lặp lại cho đến khi biến điều khiển lớn hơn giá trị cuối thì vòng lặp dừng và cho kết quả. Như vậy hết vòng lặp **for** số bước lặp là giá trị cuối (của biến điều khiển) trừ giá trị đầu cộng một.

Thí dụ: Tìm giá trị lớn nhất của dãy số $a_1, a_2, \dots, a_n$.

Thuật toán: Đầu tiên cho giá trị lớn nhất (**max**) bằng a_1 , sau đó lần lượt so sánh **max** với các số a_i ($i = 2, 3, \dots, n$), nếu $\text{max} < a_i$ thì max bằng a_i , nếu $\text{max} > a_i$ thì max không đổi.

```
Function max_day_so;  
Input:   Dãy số  $a_1, a_2, \dots, a_n$ ;  
Output:  Giá trị lớn nhất (max) của dãy số đã nhập;  
begin  
    max :=  $a_1$  ;  
    for i:= 2 to n do  
        if  $a_i > \text{max}$  then max :=  $a_i$  ;  
    max_day_so := max;  
end;
```

Chú thích: Vòng lặp **for** còn cách viết lùi biến điều khiển như sau:

for biến điều khiển := giá trị cuối **downto** giá trị đầu **do** câu lệnh hoặc khối câu lệnh;

Việc thực hiện câu lệnh này tương tự như khi viết biến điều khiển tăng dần.

b. Loại có bước lặp không xác định: Có hai cách viết

Cách thứ nhất: **while** điều kiện **do** câu lệnh hoặc khối câu lệnh;

Khi thực hiện, điều kiện được kiểm tra, nếu điều kiện được thoả mãn thì câu lệnh (khối câu lệnh) được thực hiện. Nếu điều kiện không thoả mãn thì vòng lặp dừng và cho kết quả.

Thí dụ: Kiểm tra xem số nguyên dương m đã cho có phải là số nguyên tố không?

Thuật toán như sau: Số m là số nguyên tố nếu nó không chia hết cho bất cứ số nguyên dương khác 1 nào nhỏ hơn hoặc bằng $\sqrt{m}$.

Thật vậy, nếu m là một hợp số (không phải là số nguyên tố), nghĩa là tồn tại các số nguyên dương a, b sao cho:

$$m = a.b \Rightarrow a \leq \sqrt{m} \text{ hoặc } b \leq \sqrt{m}$$

Vậy, nếu m là số nguyên tố thì m không chia hết cho mọi số nguyên dương i , $2 \leq i \leq \sqrt{m}$