

## Mục lục:

<b>1. Mối quan hệ giữa cấu trúc dữ liệu và giải thuật.</b>	<b>2</b>
1.1. Giải thuật.	2
1.2. Cấu trúc dữ liệu (CTDL).	2
1.3. Cấu trúc lưu trữ (CTLT).	3
1.4. Mối quan hệ.	4
<b>2. Các cách diễn đạt giải thuật.</b>	<b>4</b>
2.1. Bằng lời.	4
2.2. Dùng lưu đồ giải thuật.	4
2.3. Giả mã.	4
<b>3. Thiết kế và phân tích giải thuật:</b>	<b>8</b>
<b>3.1. Thiết kế giải thuật:</b>	<b>8</b>
3.1.1. <i>Modun hóa:</i>	8
3.1.2. <i>Tính chỉnh dần từng bước:</i>	8
<b>3.2. Phân tích giải thuật:</b>	<b>8</b>
3.2.1. Đặt vấn đề.	8
3.2.2. Phân tích thời gian thực hiện giải thuật:	8
3.2.3. Độ phức tạp tính toán của giải thuật.	9
<b>4. Giải thuật đệ quy.</b>	<b>10</b>
4.1. <i>Khái niệm đệ quy.</i>	10
4.2. <i>Giải thuật đệ quy và thủ tục đệ quy.</i>	10
4.3. <i>Đặc điểm</i>	11
4.4. <i>Một bài toán có những đặc điểm, tính chất gì thì thực hiện = đệ quy: ..</i>	11
4.5. <i>Ưu/nhược điểm:</i>	11
4.6. <i>Khử đệ quy.</i>	11
<b>CÂU HỎI ÔN TẬP CHƯƠNG 1:</b>	<b>12</b>

# ĐỀ CƯƠNG CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

## CHƯƠNG 1: Cấu trúc dữ liệu và giải thuật.

### 1. Mối quan hệ giữa cấu trúc dữ liệu và giải thuật.

#### 1.1. Giải thuật.

- *Khái niệm:* Giải thuật là hệ thống các thao tác, các phép toán được thực hiện theo trình tự nhất định trên một số đối tượng dữ liệu nào đó, sao cho sau một số bước hữu hạn, ta có được kết quả mong muốn.
- *Tính chất:*
  - *Tính thực hiện được:* Các thao tác, phép toán trên giải thuật mà máy tính thực hiện được.
  - *Tính kết thúc:* Giải thuật phải kết thúc sau một số bước hữu hạn và trong một thời gian hữu hạn.
  - *Tính kết quả:* Với tập dữ liệu hợp lý thì giải thuật phải cho kết quả mong đợi.
  - *Tính hiệu quả:* Trước 1 bài toán có thể có nhiều giải thuật khác nhau. Một giải thuật hiệu quả phải đơn giản, dễ hiểu, tiết kiệm thời gian và bộ nhớ.
  - *Tính duy nhất:* Cùng 1 dữ liệu đầu vào thì dù người hay máy tính thực hiện thì cũng chỉ cho kết quả duy nhất.
  - *Tính tổng quát:* Giải thuật phải áp dụng được cho mọi bài toán cùng loại.
  - *Tính hình thức:* Giải thuật yêu cầu người hay máy thực hiện, phải đúng thao tác, phép toán, không được suy diễn, thay đổi.

#### 1.2. Cấu trúc dữ liệu (CTDL).

- Giải thuật là phép xử lý, đối tượng xử lý là dữ liệu.
- Dữ liệu có 3 loại:

- Dữ liệu vào: Là dữ liệu cần đưa vào để xử lý, đây chính là đầu vào của bài toán.
- Dữ liệu trung gian: Là dữ liệu chứa các kết quả trung gian trong quá trình xử lý.
- Dữ liệu ra: Là dữ liệu chứa kết quả mong muốn của bài toán.
- Trong các loại dữ liệu thì có dữ liệu nguyên tử là nhỏ nhất, không thể phân chia được nữa. Từ các dữ liệu nguyên tử, ta tạo thành các CTDL bằng cách liên kết lại với nhau.
- *Vậy, cấu trúc dữ liệu* là tập hợp các phần tử dữ liệu liên kết với nhau bằng cách nào đó. Nói đến CTDL là nói đến cách tổ chức các phần tử dữ liệu như thế nào.

### 1.3. Cấu trúc lưu trữ (CTLT).

- Khái niệm: Là cách biểu diễn một CTDL trong bộ nhớ.
- Có 2 loại CTLT:
  - Kế tiếp:
    - Các phần tử dữ liệu được lưu trữ trong các ô nhớ liên kề nhau.
    - Mỗi ô nhớ có 1 chỉ số để truy nhập, cho phép truy nhập trực tiếp các phần tử dữ liệu
    - Vẽ hình (lưu trữ bằng mảng).
  - Phân tán:
    - Các phần tử dữ liệu được lưu trữ trong các ô nhớ nằm rải rác khắp nơi trong bộ nhớ.
    - Giữa các ô nhớ này có liên kết với nhau về địa chỉ và được gọi là nút nhớ. Trong mỗi nút nhớ ngoài phần tử dữ liệu còn có thông tin về địa chỉ nút lân cận.
    - Vẽ hình (lưu trữ phân tán).
    - Các nút nhớ được tạo ra thông qua cơ chế cấp phát bộ nhớ động.

#### 1.4. Mối quan hệ.

##### - CTDL ~ CTLT:

- 1 CTDL có thể sử dụng nhiều CTLT.
  - VD: Ngăn xếp được lưu trữ bằng CTLT kế tiếp hoặc phân tán.
- 1 CTLT có thể dùng cho nhiều CTDL.
  - VD: CTLT kế tiếp được sử dụng để lưu trữ ngăn xếp và hàng đợi.

##### - CTDL ~ Giải thuật:

- CTDL nào thì giải thuật đó. Khi CTDL thay đổi thì giải thuật thay đổi theo.
- Mối quan hệ giữa CTDL và giải thuật được thể hiện bằng phương trình sau: **CTDL + Giải thuật = Chương trình**

## 2. Các cách diễn đạt giải thuật.

### 2.1. Bảng lời.

- Sử dụng ngôn ngữ tự nhiên để liệt kê tuần tự từng bước một.
- Ưu điểm: Dễ diễn đạt, dễ hiểu.
- Nhược điểm: Nếu diễn đạt không tốt thì gây khó hiểu cho người lập trình.

### 2.2. Dùng lưu đồ giải thuật.

- Là 1 đồ thị có hướng.
- Các hình cơ bản.
- Ưu điểm: dễ diễn đạt.
- Nhược điểm: Nếu giải thuật lớn thì gây không khềnh.

### 2.3. Giả mã.

- Ta sử dụng kết hợp ngôn ngữ tự nhiên và 1 số từ khóa, cấu trúc lệnh của 1 ngôn ngữ lập trình bậc cao nào đó để diễn đạt.

#### a. Quy định chung:

- Tên chương trình viết bằng chữ hoa, có thể thêm dấu gạch ngang và đặt sau từ khóa Program.
- Lời chú thích được đặt giữa dấu ngoặc {...}.

b. Biểu thức:

- Phép toán:
  - o Số học: +, -, \*, /, ^, DIV, MOD.
  - o Quan hệ: <, >, ≤, ≥, ≠, =.
  - o Logic: NOT, AND, OR, XOR.
  - o Các giá trị logic: True, False.

- Tên biến: Theo quy tắc đặt tên biến trong ngôn ngữ lập trình.
- Biến chỉ số: Tên[chỉ số].
- Biểu thức: đặt trong dấu (biểu thức).

c. Câu lệnh:

- Thể hiện các thao tác, công việc, kết thúc bằng dấu chấm phẩy.
- Phép gán: := hoặc ←.
- Phép hoán đổi: :=: hoặc ↔.
- Liệt kê cấu trúc tuần tự: 1, 2, ...
- Câu lệnh ghép: Muốn ghép lại thành 1 khối lệnh.

Begin

S1; S2;...

End;

- Câu lệnh điều kiện:
  - o If ... then ...
  - o If ... then ... else ...
- Câu lệnh lựa chọn:

Case

B1: S1;

B2: S2;

...

Else  $S_{n+1}$ ;

End case

- Câu lệnh lặp:
    - Lặp với số lần biết trước:
      - For  $i:=m$  to  $n$  do S;
      - For  $i:=n$  downto  $m$  do S;
    - Lặp với số lần không biết trước:
      - While B do S;
      - Repeat S until B;
  - Câu lệnh chuyển: goto n;
  - Câu lệnh vào ra:
    - Read(danh sách biến);
    - Write(danh sách hằng, biến, biểu thức);
  - Câu lệnh kết thúc: end.
- d. Chương trình con: Giả mã Pascal được tổ chức thành 3 dạng:
- \*Cấu trúc giả mã dạng chương trình:**
- Vào:
  - Ra:
- {Mô tả}

Program TÊN\_CHƯƠNG\_TRÌNH

1.

2.

...

End.

**\*Cấu trúc giả mã dạng hàm:**

- Vào:
- Ra:
- {Mô tả}

Function Tên\_hàm(danh sách tham số)

S1;

S2;

...

Tên\_Hàm:=biểu thức;

Return

**\*Cấu trúc giả mã dạng thủ tục:**

- Vào:
- Ra:
- {Mô tả}

Procedure Tên\_thủ\_tục(danh sách tham số)

S1;

S2;

...

Return

e. Lời gọi chương trình con:

- Dạng hàm: **Tên\_hàm(danh sách tham số thực)**
- Dạng thủ tục: **Call Tên\_thủ\_tục(danh sách tham số thực)**

### 3. Thiết kế và phân tích giải thuật:

#### 3.1. Thiết kế giải thuật:

- Trước 1 bài toán, để xây dựng giải thuật, ta trải qua 2 giai đoạn:
  - o GD 1: Thiết kế tổng quan, ta sử dụng phương pháp *modun hóa*.
  - o GD 2: Thiết kế chi tiết, sử dụng phương pháp *tinh chỉnh dần từng bước*.

##### 3.1.1. Modun hóa:

- Nội dung:
 

Coi bài toán ban đầu như 1 modun chính và phân chia nó thành các modun con. Mỗi modun con lại phân chia tiếp cho tới khi mỗi modun con đó ứng với các phần việc cơ bản mà ta đã biết hướng giải.
- Mô hình: (Vẽ hình).

##### 3.1.2. Tinh chỉnh dần từng bước:

- Nội dung:
 

Đầu tiên trình bày giải thuật bằng ngôn ngữ tự nhiên để phản ánh ý chính. Sau đó chi tiết dần dần, tương ứng với các công việc nhỏ hơn, gọi là bước tinh chỉnh. Càng ở các bước sau thì công việc được mô tả hướng tới các lệnh của giả mã.
- Mô hình: **Ngôn ngữ tự nhiên → Các bước tinh chỉnh → Giả mã.**

#### 3.2. Phân tích giải thuật:

##### 3.2.1. Đặt vấn đề.

##### 3.2.2. Phân tích thời gian thực hiện giải thuật:



- Phân tích thời gian: Thời gian thực hiện giải thuật là tiêu chuẩn đánh giá hiệu lực của giải thuật.
  - o Với một bài toán có nhiều giải thuật, ta cần chọn giải thuật dẫn đến kết quả nhanh nhất.
  - o Thời gian thực hiện phụ thuộc vào nhiều yếu tố như:
    - Kích thước của dữ liệu vào.
    - Các kiểu lệnh, tốc độ xử lý của máy tính, ngôn ngữ viết chương trình, chương trình dịch cũng ảnh hưởng đến tốc độ thực hiện.

### 3.2.3. Độ phức tạp tính toán của giải thuật.

\*Độ phức tạp tính toán của giải thuật:

- Cách đánh giá thời gian thực hiện giải thuật không phụ thuộc vào máy tính và các yếu tố liên quan mà chỉ phụ thuộc vào kích thước dữ liệu đầu vào được gọi là đánh giá theo “Độ phức tạp tính toán của giải thuật”.
- Nếu thời gian thực hiện một giải thuật là  $T(n) = Cn^2$ , trong đó  $C$  là hằng số, thì ta nói độ phức tạp tính toán của giải thuật này có cấp  $n^2$ , và được kí hiệu là:  $T(n) = O(n^2)$
- Tổng quát: Hàm  $f(n)$  có độ phức tạp tính toán cấp  $g(n)$ , kí hiệu là  $f(n) = O(g(n))$ , nếu tồn tại các hằng số  $C$  và  $n_0$  sao cho:

$$f(n) \leq Cg(n) \text{ với } n \geq n_0$$

nghĩa là hàm  $f(n)$  bị chặn trên bởi  $Cg(n)$ , với  $C$  là hằng số và với mọi  $n$  từ một điểm nào đó.

\*Xác định độ phức tạp tính toán của giải thuật.

1) Quy tắc cộng:

- Giả sử  $T1(n)$  và  $T2(n)$  là thời gian thực hiện 2 đoạn chương trình P1 và P2 mà  $T1(n) = O(f(n))$ ,  $T2(n) = O(g(n))$ , thì thời gian thực hiện P1 rồi đến P2 tiếp theo sẽ là:  $T1(n) + T2(n) = O(\max(f(n), g(n)))$

### 2) Quy tắc nhân:

- Nếu tương ứng với 2 bước P1 và P2 là  $T1(n) = O(f(n))$ ,  $T2(n) = O(g(n))$  thì thời gian thực hiện P1 và P2 lồng nhau là:  $T1(n).T2(n) = O(f(n).g(n))$

### 3) Quy tắc bỏ hằng số

- $O(c.f(n)) = O(f(n))$ , trong đó  $c$  là một hằng số.

“Khi đánh giá thời gian thực hiện giải thuật ta chỉ cần chú ý tới các bước tương ứng với một phép toán được gọi là phép toán tích cực. Đó là phép toán mà thời gian thực hiện nó không ít hơn thời gian thực hiện các phép toán khác.”

### Các bước xác định ô lớn của 1 giải thuật:

B1. Xác định phép toán tích cực trong giải thuật. (phép toán có thời gian thực hiện là lớn nhất)

B2. Đếm số lần thực hiện phép toán tích cực.

B3. Biểu diễn số lượng đếm được thành một hàm  $f(n)$  phụ thuộc vào kích thước dữ liệu đầu vào.

B4. Áp dụng các quy tắc để tìm  $O$ .

## 4. Giải thuật đệ quy.

### 4.1. Khái niệm đệ quy.

- Ta nói một đối tượng là đệ quy nếu nó được định nghĩa dưới dạng chính nó.
- Ví dụ:

Hàm  $n!$

- $n! = n * (n-1)!$  nếu  $n > 0$ .
- $n! = 1$  nếu  $n = 0$ .

### 4.2. Giải thuật đệ quy và thủ tục đệ quy.

- Nếu lời giải của một bài toán  $T$  được thực hiện bằng lời giải của bài toán  $T'$  có dạng giống như  $T$  thì đó là một *lời giải đệ quy*. Trong đó  $T'$  tuy giống  $T$  nhưng nó phải nhỏ hơn  $T$ .
- Giải thuật tương ứng với lời giải đệ quy gọi là giải thuật đệ quy.
- Thủ tục viết cho bài toán có lời giải đệ quy gọi là thủ tục đệ quy.
- Trong thủ tục đệ quy có lời gọi tới chính nó, mỗi lần gọi thì kích thước bài toán thu nhỏ hơn và dần dần tiến tới trường hợp đặc biệt là trường hợp suy biến.

(ví dụ xem trong bài giảng)

#### 4.3. Đặc điểm

- Luôn có lời gọi tới hàm đến chính nó.
- Trong thủ tục đệ quy luôn có lệnh if để kiểm tra điều kiện dừng (dừng đệ quy) đây chính là trường hợp suy biến.

#### 4.4. Một bài toán có những đặc điểm, tính chất gì thì thực hiện = đệ quy:

- Có thể định nghĩa dưới dạng 1 bài toán cùng loại nhưng kích thước nhỏ hơn
- Phải có TH đặc biệt (suy biến) để kết thúc bài toán
- Mỗi lần thực hiện bài toán thì kích thước giảm đi

#### 4.5. Ưu và nhược điểm:

- Ưu: Ngắn gọn, sáng sủa
  - Nhược: Thời gian ra vào chương trình con
- ➔ Tốn thời gian và bộ nhớ
- ➔ Không nên dùng đệ quy khi có 1 giải thuật khác

#### 4.6. Khử đệ quy

- Tìm lời gọi đệ quy có thể thay bằng vòng lặp
- Điều kiện dừng đệ quy cũng chính là điều kiện dừng vòng lặp hay xét Th đặc biệt.

**CÂU HỎI ÔN TẬP CHƯƠNG 1:**

**Câu 1:** Nêu khái niệm cấu trúc dữ liệu, giải thuật, mối quan hệ giữa cấu trúc dữ liệu với giải thuật.

**Câu 2:** Nêu khái niệm giải thuật, các tính chất của giải thuật.

**Câu 3:** Nêu khái niệm cấu trúc lưu trữ, mối quan hệ giữa cấu trúc dữ liệu và cấu trúc lưu trữ.

**Câu 4:** Nêu khái niệm giải thuật. Các cách diễn đạt của giải thuật.

**Câu 5:** Trình bày ý tưởng của phương pháp mô đun hóa và tính chỉnh dần từng bước trong thiết kế giải thuật.

**Câu 6:** Hãy nêu khái niệm về độ phức tạp tính toán  $O$  của giải thuật. Trình bày các quy tắc cơ bản xác định độ phức tạp tính toán của giải thuật. Xác định  $O$  cho giải thuật sau.

**Câu 7:** Hãy nêu khái niệm về độ phức tạp tính toán  $O$  của giải thuật. Liệt kê các hàm có độ phức tạp  $O$  theo thứ tự giảm dần (tăng dần).

**Câu 8:** Hãy nêu khái niệm về độ phức tạp tính toán  $O$  của giải thuật. Cho biết ý nghĩa độ phức tạp của giải thuật.

**Câu 9:** Hãy nêu khái niệm về độ phức tạp tính toán  $O$  của giải thuật. Trình bày các bước xác định  $O$ . Xác định  $O$  cho giải thuật sau.

**Câu 10:** Nêu khái niệm đệ quy, giải thuật đệ quy, thủ tục đệ quy, đặc điểm của thủ tục đệ quy, một bài toán như thế nào thì có thể thực hiện bằng giải thuật đệ quy, trình bày phương pháp khử đệ quy (áp dụng khử đệ quy cho giải thuật tìm UCLN), ưu - nhược điểm, cho bài toán, cho bài toán, thiết kế giải thuật đệ quy.