

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

03 tín chỉ (02 LT - 01 TH)

Giảng viên: Đinh Thị Lan Phương

NỘI DUNG MÔN HỌC

- Chương 1: Tổng quan về CTDL & GT
- Chương 2 : Định quy và giải thuật đệ quy
- Chương 3 : Danh sách tuyến tính
- Chương 4 : Cấu trúc cây
- Chương 5 : Các giải thuật sắp xếp
- Chương 6 : Các giải thuật tìm kiếm

TÀI LIỆU THAM KHẢO

- Cấu trúc dữ liệu và giải thuật, Đỗ Xuân Lô, NXB Thống kê.
- Cấu trúc dữ liệu và thuật toán, Đinh Mạnh Tường, NXB Khoa học kỹ thuật
- Đề cương chi tiết học phần CTDL>, An Văn Minh, Khoa CNTT, ĐHCNHN.

Chương 1- TỔNG QUAN

- Giải thuật và cấu trúc dữ liệu
- Phân tích và đánh giá giải thuật
- Các cấu trúc dữ liệu cơ sở

1.1 GIẢI THUẬT VÀ CTDL

- Giải thuật
- Cấu trúc dữ liệu
- Mối quan hệ giữa CTDL>

1.1.1 Giải thuật

- Khái niệm
- Đặc trưng của giải thuật
- Các phương pháp diễn đạt giải thuật

Khái niệm

- Giải thuật là một dãy các thao tác, được mô tả chính xác theo trình tự nhất định để giải quyết bài toán sau một số hữu hạn các bước

Đặc trưng của giải thuật

- Bộ dữ liệu vào
- Dữ liệu ra
- Tính xác định
- Tính dừng
- Tính phổ dụng
- Tính hiệu quả

Các phương pháp diễn đạt giải thuật

- Sử dụng ngôn ngữ tự nhiên
- Sử dụng sơ đồ khối
- Sử dụng ngôn ngữ lập trình

Ví dụ các phương pháp diễn đạt giải thuật

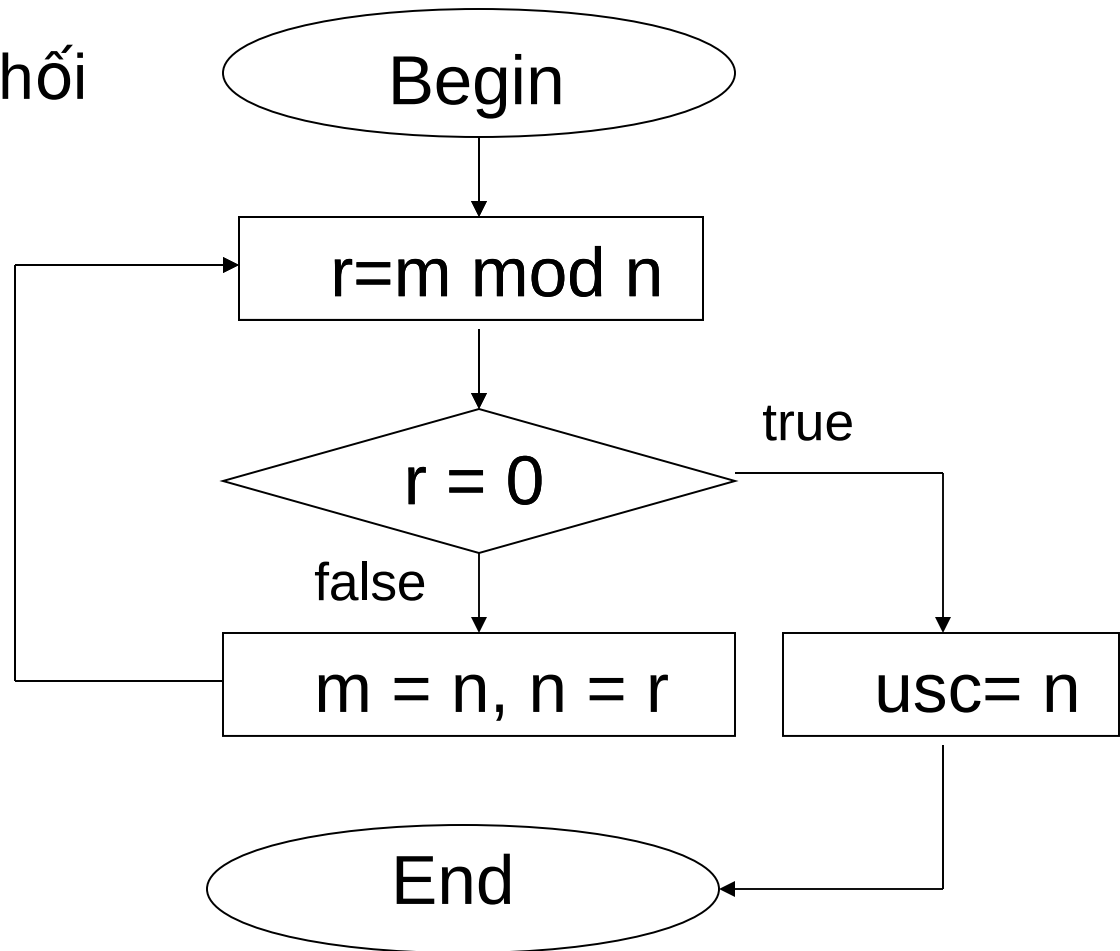
- Diễn đạt giải thuật tìm ước số chung lớn nhất của 2 số nguyên dương m, n theo thuật toán Euclid
 - *In put* : 2 số nguyên dương m, n
 - *Out put* : Ước số chung lớn nhất của 2 số m, n

Ví dụ các phương pháp diễn đạt giải thuật

- Sử dụng ngôn ngữ tự nhiên
 - Bước 1: Lấy m chia dư cho n được số dư là r
 - Bước 2: Kiểm tra r
 - Nếu $r = 0$: USCLN là n, kết thúc.
 - Nếu $r \neq 0$: Gán $m = n$, $n = r$, quay lại bước 1

Ví dụ các phương pháp diễn đạt giải thuật

- Sử dụng sơ đồ khối



Ví dụ các phương pháp diễn đạt giải thuật

- Sử dụng ngôn ngữ lập trình

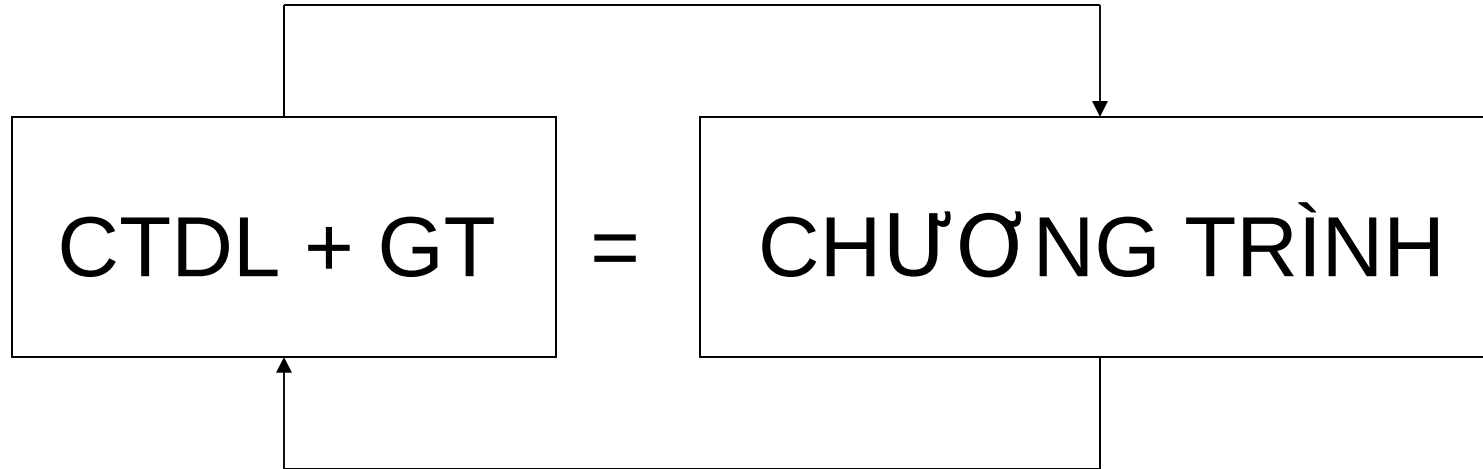
```
int Euclid (int m, int n)
{
    int r ;
    r = m % n;
    while (r != 0)
    {
        m = n;
        n = r;
        r = m % n;
    }
    return n;
}
```

1.1.2 Cấu trúc dữ liệu

- Khái niệm
 - *Là cách biểu diễn các dữ liệu dùng để mô tả các đối tượng cần xử lý trong máy tính*
- Các tiêu chuẩn đánh giá cấu trúc dữ liệu
 - *Phản ánh đúng thực tế*
 - *Phù hợp với các giải thuật xử lý trên đó*
 - *Tiết kiệm tài nguyên hệ thống*

1.1.3 Mối quan hệ giữa CTDL>

- Được thể hiện bởi sơ đồ :



1.2 PHÂN TÍCH VÀ ĐÁNH GIÁ GIẢI THUẬT

- Thời gian thực hiện của giải thuật
- Đánh giá độ phức tạp tính toán của giải thuật
- Phương pháp xác định độ phức tạp tính toán

1.2.1 Thời gian thực hiện giải thuật

- Thời gian thực hiện giải thuật trong thực tế phụ thuộc vào nhiều yếu tố :
 - Kích thước dữ liệu cần xử lí
 - Cách thức nhập dữ liệu
 - Chương trình dịch
 - Tốc độ xử lí của máy tính

1.2.1 Thời gian thực hiện giải thuật

- Cách đánh giá thời gian thực hiện giải thuật độc lập với máy tính và các yếu tố liên quan tới máy tính gọi là độ phức tạp tính toán của giải thuật
- Thời gian thực hiện giải thuật là số các phép toán cơ bản cần thực hiện thông qua độ lớn của dữ liệu
- Thời gian thực hiện giải thuật kí hiệu là $T(n)$ trong đó n là kích thước dữ liệu

1.2.2 Đánh giá độ phức tạp tính toán của giải thuật

- Định nghĩa :

- $T(n) = O(g(n))$ Nếu tồn tại các hằng số c và n_0 sao cho

$$T(n) \leq c \cdot g(n) \text{ với mọi } n \geq n_0$$

- Ví dụ :

$$T(n) = 2n^3 + 3n^2$$

Ta có : $T(n) \leq 5n^3$ với mọi $n \geq 0$

Vậy $T(n) = O(n^3)$ (Kí pháp chữ O lớn)

1.2.2 Đánh giá độ phức tạp tính toán của giải thuật

- Với một bài toán nếu :
 - $T(n) = O(1)$ Độ phức tạp hằng số
 - $T(n) = O(n)$ Độ phức tạp tuyến tính
 - $T(n) = O(n^2, n^3, \log_2(n), \dots)$
Độ phức tạp đa thức
 - $T(n) = O(a^n, n!, \dots)$ Độ phức tạp hàm mũ

1.2.3 Phương pháp đánh giá độ phức tạp giải thuật

- **Quy tắc tổng :**

Giả sử $T1(n)$ và $T2(n)$ là thời gian thực hiện của hai đoạn chương trình P1 và P2 kế tiếp nhau

mà $T1(n) = O(f(n)); T2(n) = O(g(n))$

thì $T(n) = T1(n) + T2(n) = O(\max(f(n),g(n)))$

- **Qui tắc nhân:**

Nếu $T1(n)$ và $T2(n)$ là thời gian thực hiện của hai đoạn chương trình P1 và P2 lồng nhau

thì $T(n) = T1(n).T2(n) = O(f(n).g(n))$

1.2.3 Phương pháp đánh giá độ phức tạp giải thuật

- Đánh giá độ phức tạp của các lệnh trong chương trình
 - Câu lệnh gán, lời gọi nhập xuất dữ liệu : $O(1)$
 - Câu lệnh rẽ nhánh, lựa chọn : Theo quy tắc tổng
 - Câu lệnh lặp : Theo quy tắc nhân

1.2.3 Phương pháp đánh giá độ phức tạp giải thuật

- Chú ý khi đánh giá độ phức tạp của giải thuật
 - Quan tâm tới các phép toán tích cực
 - Xác định độ phức tạp của giải thuật thông qua tình trạng xấu nhất của dữ liệu
- Ví dụ :
 - Đánh giá độ phức tạp của giải thuật tính giá trị của e^x theo công thức

$$e^x = 1 + x/1! + x^2/2! + \dots + x^n/n!, \text{ với } x \text{ và } n \text{ cho trước}$$

1.2.3 Phương pháp đánh giá độ phức tạp giải thuật

```
float Exp1(int n, float x)
{ //Tính từng số hạng sau đó cộng dồn lại
    float s, p;
    int i, j;
        s = 1;                               (1)
    for (i =1; i<=n; i++)                     (2)
    { p = 1;                                   (3)
        for (j =1; j<=i; j++)                 (4)
            p = p*x/j;                         (5)
        s = s + p;                             (6)
    }
    return s;                                 (7)
}
```


1.2.3 Phương pháp đánh giá độ phức tạp giải thuật

```
float Exp2 (int n, float x)
{ float s, p;
  int i;
  s = 1; (1)
  p = 1; (2)
  for (i =1; i<= n; i++) (3)
    { p = p*x/i;; (4)
      s = s + p; (5)
    }
  return s; (6)
}
```

1.3 CÁC CTDL CƠ SỞ

- Dữ liệu kiểu đơn giản chuẩn
- Dữ liệu kiểu có cấu trúc
- Dữ liệu kiểu con trỏ
- Các thuộc tính của dữ liệu

1.3.1 Dữ liệu kiểu đơn giản chuẩn

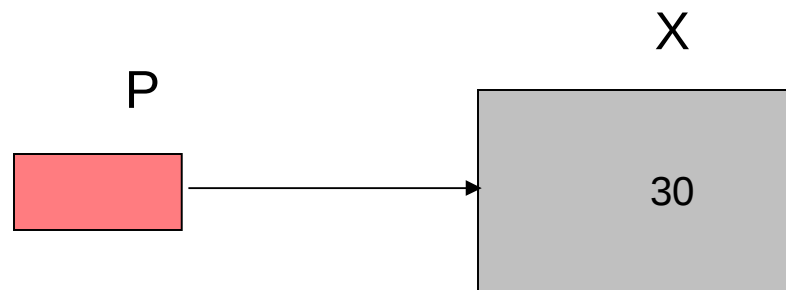
- Dữ liệu kiểu nguyên
- Dữ liệu kiểu thực
- Dữ liệu kiểu logic
- Dữ liệu kiểu kí tự

1.3.1 Dữ liệu kiểu có cấu trúc

- Dữ liệu kiểu mảng
- Dữ liệu kiểu xâu
- Dữ liệu kiểu bản ghi
- Dữ liệu kiểu tập hợp
- Dữ liệu kiểu tệp

1.3.3 Dữ liệu kiểu con trỏ

- Con trỏ là biến được sử dụng để lưu địa chỉ của một biến khác.
- Con trỏ được khai báo theo mẫu:
<kiểu dữ liệu> *<tên con trỏ> ;
- Ví dụ : *int *P, x ;*
P=&x; // P chứa địa chỉ của biến x, hay P trỏ vào x



1.3.4 Các thuộc tính của dữ liệu

- Tên kiểu
- Miền giá trị
- Kích thước lưu trữ
- Tập các toán tử tác động lên kiểu dữ liệu
 - Truy nhập dữ liệu
 - Kết hợp dữ liệu
 - Các phép toán số học : +, -, *, /, %, ^, ...
 - Các phép toán logic : &&, ||, !
 - Các phép toán quan hệ : >, <, >=, <=, =, <>

BÀI TẬP ÁP DỤNG

- Xây dựng và đánh giá độ phức tạp của các giải thuật sau trong mảng một chiều n phần tử là các số nguyên :
 - Tìm số lớn nhất và số phần tử lớn nhất
 - Loại các phần tử bằng x ra khỏi mảng
 - Sắp xếp lại các phần tử theo thứ tự tăng dần
 - Ghép 2 mảng đã xếp vào 1 mảng được xếp