CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

Giảng viên: ThS. PHAN NGUYỆT THUẦN

Bộ môn Vật lý Tin học – Khoa Vật lý – Vật lý Kỹ thuật-ĐH KHTN TPHCM

Email: pnthuan@hcmuns.edu.vn

Tài Liệu Tham Khảo

- Cấu trúc dữ liệu và giải thuật, Đỗ Xuân Lôi, NXB ĐHQG Hà Nội
- Bài giảng CTDL, Phạm Thế Bảo, ĐH KHTN TPHCM
- Bài giảng CTDL, ĐH Công nghệ thông tin TPHCM
- Bài giảng CTDL, Hồ Sĩ Đàm, ĐH Công nghệ, ĐHQG Hà Nội
- Thiết kế và đánh giá thuật toán, Trần Tuấn Minh, ĐH Đà Lạt
- Nhập môn CTDL, Dương Anh Đức, Trần Hạnh Nhi, ĐH KHTN TPHCM

Giới thiệu môn học

Mục tiêu

- Giới thiệu các CTDL cơ bản
- Trình bày các phương pháp tìm kiếm và sắp xếp nội
- Trình bày các cấu trúc xâu và các thao tác trên xâu
- Trình bày các cấu trúc cây và thao tác trên cây
- Cài đặt minh họa: Ngôn ngữ C

Kiến thức tiên quyết

Kỹ thuật lập trình C

CHƯƠNG 1

TỔNG QUAN VỀ CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

Nội Dung

- Tổng quan về CTDL và thuật toán
- Các tiêu chuẩn của CTDL
- Vai trò của CTDL
- Độ phức tạp của thuật toán
- Thực hiện và hiệu chỉnh chương trình
- Tiêu chuẩn của chương trình

Khái Niệm Về CTDL Và Thuật Toán

- Niklaus Wirth:

  CTDL + Thuật toán = Chương trình

- Cần nghiên cứu về thuật toán và CTDL!

Sự Cần Thiết Của Thuật Toán

- Tại sao sử dụng máy tính để xử lý dữ liệu?
  - Nhanh hơn.
  - Nhiều hơn.
  - Giải quyết những bài toán mà con người không thể hoàn thành được.
- Làm sao đạt được những mục tiêu đó?
  - Nhờ vào sự tiến bộ của kỹ thuật: tăng cấu hình máy ⇨ chi phí cao ☹
  - Nhờ vào các thuật toán hiệu quả: thông minh và chi phí thấp ☺

Thuật Toán

- **Thuật toán**: Một dãy hữu hạn các chỉ thị có thể thi hành để đạt mục tiêu đề ra nào đó.
- **Ví dụ**: Thuật toán tính tổng tất cả các số nguyên dương nhỏ hơn n gồm các bước sau:

Bước 1: *S=0, i=1;*

Bước 2: *nếu i<n thì s=s+i;*

Ngược lại: qua bước 4;

Bước 3:

i=i+1;

Quay lại bước 2;

Bước 4: *Tổng cần tìm là S.*

Các Tiêu Chuẩn Của Thuật Toán

- Xác định
- Hữu hạn
- Đúng
- Tính hiệu quả
- Tính tổng quát

Biểu Diễn Thuật Toán

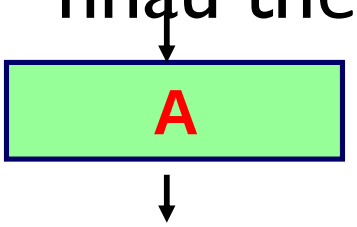
- Dạng ngôn ngữ tự nhiên
- Dạng lưu đồ (sơ đồ khối)
- Dạng mã giả
- Ngôn ngữ lập trình

Biểu Diễn Bằng Ngôn Ngữ Tự Nhiên

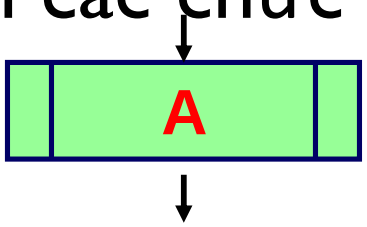
- NN tự nhiên thông qua các bước được tuần tự liệt kê để biểu diễn thuật toán.
- Ưu điểm:
  - Đơn giản, không cần kiến thức về về cách biểu diễn (mã giả, lưu đồ,...)
- Nhược điểm:
  - Dài dòng, không cấu trúc.
  - Đôi lúc khó hiểu, không diễn đạt được thuật toán.

Lưu Đồ

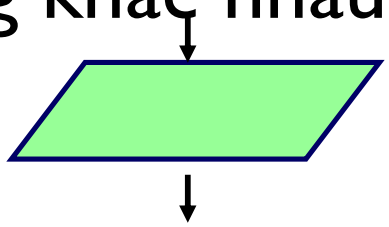
- Là hệ thống các nút, cung hình dạng khác nhau thể hiện các chức năng khác nhau.



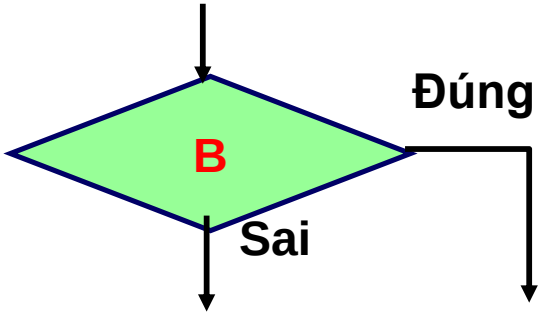
Thực hiện A



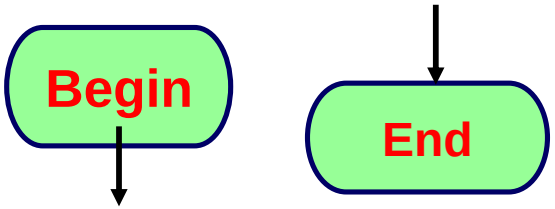
Gọi hàm A



Vào / Ra dữ liệu

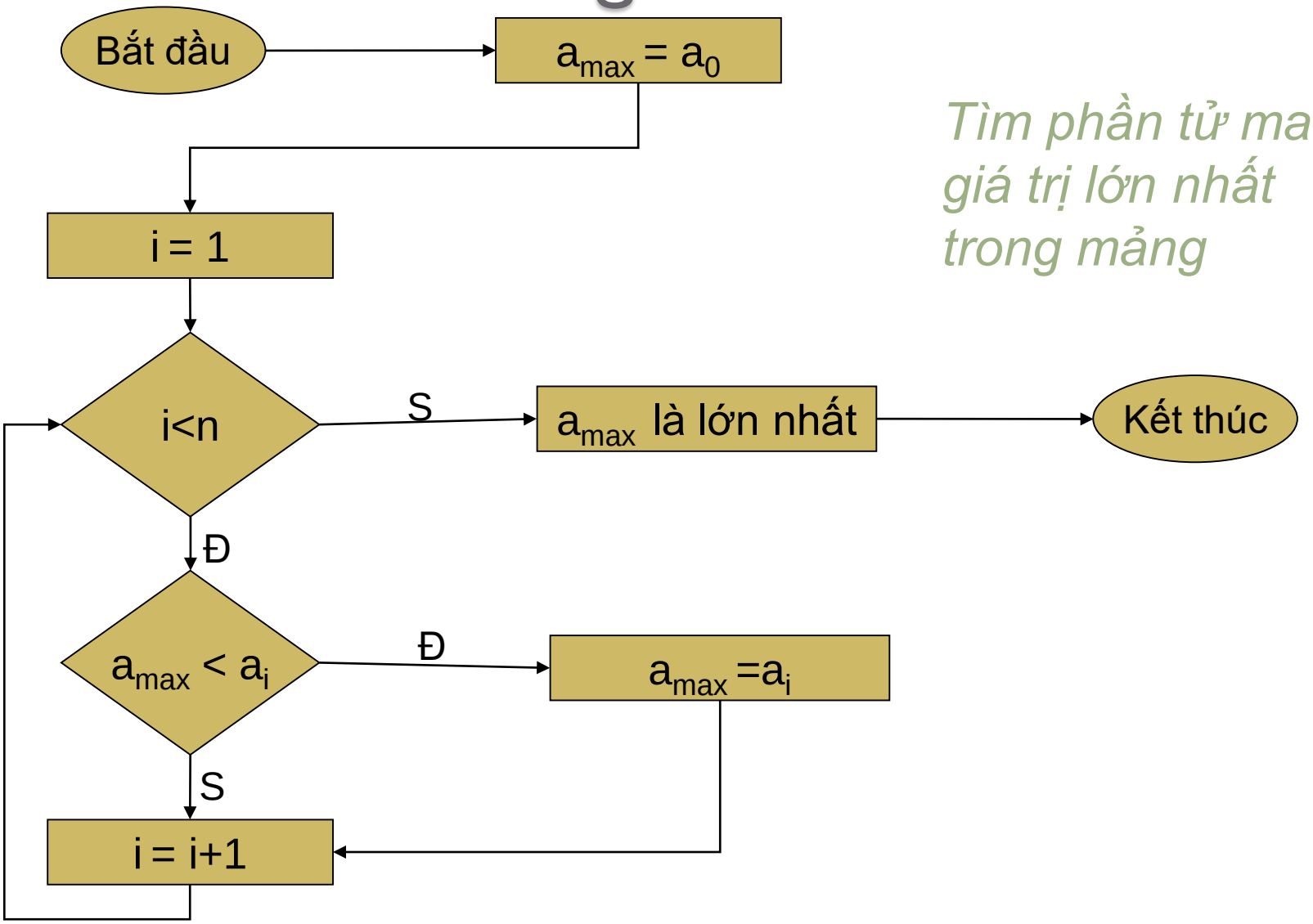


Điều kiện rẻ nhánh B



Nút giới hạn bắt đầu / kết thúc chương trình

Biểu Diễn Bằng Lưu Đồ



Tìm phần tử mang giá trị lớn nhất trong mảng

Biểu Diễn Bằng Mã Giả

- Ngôn ngữ tựa ngôn ngữ lập trình:
  - Dùng cấu trúc chuẩn hóa, chẳng hạn tựa Pascal, C.
  - Dùng các ký hiệu toán học, biến, hàm.
- Ưu điểm:
  - Đỡ cồng kềnh hơn lưu đồ khối.
- Nhược điểm:
  - Không trực quan bằng lưu đồ khối.

Biểu Diễn Bằng Mã Giả

- **Một số quy ước**
  1. Các biểu thức toán học
  2. Lệnh gán: "=" (A←B)
  3. So sánh: "==", "!="
  4. Khai báo hàm (thuật toán)

Thuật toán *<tên TT> (<tham số>)*

Input: *<dữ liệu vào>*

Output: *<dữ liệu ra>*

<Các câu lệnh>

End

Biểu Diễn Bằng Mã Giả

5. Các cấu trúc:

Cấu trúc chọn:

if … **then** … [**else** …]

Vòng lặp:

while … **do**

do … **while** (…)

for … **do** …

6. Một số câu lệnh khác:

Trả giá trị về: **return** [giá trị]

Lời gọi hàm: <Tên>(tham số)

Biểu Diễn Bằng Mã Giả

- **Ví dụ**: Tìm phần tử lớn nhất trong mảng một chiều.

$a_{max}=a_0$;
i=1;
while (i<n)
  if ($a_{max}<a_i$) $a_{max} = a_i$;
    i++;
end while;

Biểu Diễn Bằng Ngôn Ngữ Lập Trình

- Dùng ngôn ngữ máy tính (C, Pascal,...) để diễn tả thuật toán, CTDL thành câu lệnh.
- Dùng phương pháp tinh chế từng bước để chuyển hoá bài toán sang mã chương trình cụ thể.

Độ Phức Tạp Của Thuật Toán

- Một thuật toán hiệu quả:
  - Chi phí cần sử dụng tài nguyên thấp: Bộ nhớ, thời gian sử dụng CPU, …
- Phân tích độ phức tạp thuật toán:
  - **N** là khối lượng dữ liệu cần xử lý.
  - Mô tả độ phức tạp thuật toán qua một hàm **f(N)**.
  - Hai phương pháp đánh giá độ phức tạp của thuật toán:
    - Phương pháp thực nghiệm.
    - Phương pháp xấp xỉ toán học.

Phương Pháp Thực Nghiệm

- Cài thuật toán rồi chọn các bộ dữ liệu thử nghiệm.
- Thống kê các thông số nhận được khi chạy các bộ dữ liệu đó.
- *Ưu điểm*: Dễ thực hiện.
- *Nhược điểm*:
  - Chịu sự hạn chế của ngôn ngữ lập trình.
  - Ảnh hưởng bởi trình độ của người lập trình.
  - Chọn được các bộ dữ liệu thử đặc trưng cho tất cả tập các dữ liệu vào của thuật toán: khó khăn và tốn nhiều chi phí.
  - Phụ thuộc vào phần cứng.

Phương Pháp Xấp Xỉ

- Đánh giá giá thuật toán theo hướng tiệm xấp xỉ tiệm cận qua các khái niệm O().
- *Ưu điểm*: Ít phụ thuộc môi trường cũng như phần cứng hơn.
- *Nhược điểm*: Phức tạp.
- Các trường hợp độ phức tạp quan tâm:
  - Trường hợp tốt nhất (phân tích chính xác)
  - Trường hợp xấu nhất (phân tích chính xác)
  - Trường hợp trung bình (mang tích dự đoán)

Sự Phân Lớp Theo Độ Phức Tạp Của Thuật Toán

- **Sử dụng ký hiệu BigO**
  - Hằng số: O(c)
  - logN : O(logN)
  - N : O(N)
  - NlogN : O(NlogN)
  - N^2 : O(N^2)
  - N^3 : O(N^3)
  - 2^N : O(2^N)
  - N! :O(N!)

Độ phức tạp tăng dần

Quy tắc xác định độ phức tạp

- Quy tắc tổng:
  - P1, T1(n)=O(f(n))
  - P2, T2(n)=O(g(n))

  ➔T1(n) + T2(n) = O((f(n)+g(n)))
- Quy tắc nhân:
  - P1, P2 lồng nhau

  ➔T1(n)T2(n) = O(f(n)g(n))

Quy tắc xác định độ phức tạp

- Quy tắc Max

  Nếu P có T(n)= O(f(n)+g(n))

thì P có độ phức tạp là O(max (f(n), g(n))).

- Quy tắc xác định O(?): xét thành phần có bậc cao nhất của f
  - $12n - 2 \in O(n)$
  - $3n^2\log(n) - 12n^2 + 19 \in O(n^2\log(n))$

Áp dụng đánh giá chương trình

Câu lệnh đơn thực hiện một thao tác

QT hằng số

Câu lệnh hợp thành là dãy các câu lệnh

QT tổng

Câu lệnh rẽ nhánh dạng If ..then..else.

QT Max

- Các câu lệnh lặp

QT Nhân

Một số lớp thuật toán

n \ Hàm	n	lg n	Nlgn	n^2	n^3	2^n
1	1	0	0	1	1	2
2	2	1	2	4	8	4
4	n	2	8	16	64	16
8	8	3	24	64	512	256
16	16	4	64	256	4096	65536
32	32	5	160	1024	32768	2,147,483,648

Cấu Trúc Dữ Liệu

- Cách tổ chức lưu trữ dữ liệu.
- Các tiêu chuẩn của CTDL:
  - Phải biểu diễn đầy đủ thông tin.
  - Phải phù hợp với các thao tác trên đó.
  - Tiết kiệm tài nguyên hệ thống.

Vai Trò Của Cấu Trúc Dữ Liệu

- Cấu trúc dữ liệu đóng vai trò quan trọng trong việc kết hợp và đưa ra cách giải quyết bài toán.
- CTDL hỗ trợ cho các thuật toán thao tác trên đối tượng được hiệu quả hơn

Khái niệm kiểu dữ liệu

T = <V, O>

V = {Tập các giá trị}

O = {Tập các thao tác xử lý được phép thực hiện}

Ví dụ: Kiểu dữ liệu số nguyên int trong ngôn ngữ C

T = int

V = {-32768, 32767}

O = {+, -, *, /, %}

Các thuộc tính của một kiểu dữ liệu

- Tên.
- Miền giá trị.
- Kích thước lưu trữ.
- Tập các thao tác tác động lên kiểu dữ liệu đó

Các loại kiểu dữ liệu

- Kiểu dữ liệu cơ bản: Cơ sở, mảng, cấu trúc cơ bản.
- Kiểu dữ liệu có cấu trúc hướng giải quyết vấn đề: Danh sách liên kết, hàng đợi, ngăn xếp, cây, bảng băm, …

Minh họa chương trình quản lý sách đơn giản trong thư viện. Sử dụng cấu trúc dữ liệu danh sách liên kết đơn để cài đặt danh sách chứa nội dung các cuốn sách.

A. Thông tin liên quan đến một cuốn sách gồm:

- Mã số sách
- Tên sách
- Tên tác giả
- Nhà xuất bản
- Năm xuất bản
- Trạng thái sách: {TRUE: chưa mượn/ FALSE: đã mượn}

B. Các thao tác trên danh sách này:

1. Khởi tạo danh sách
    a. Khởi tạo danh mục sách rỗng (chưa có sách)
    b. Đọc từ file: nhập vào tên file đã lưu danh mục sách ở lần làm việc trước đó.
2. Thêm một cuốn sách vào danh sách
    a. Thêm vào đầu danh sách: InsertFirst
    b. Thêm vào sau một cuốn sách nào đó: InsertAfter
    c. Thêm vào cuối của danh sách: InsertLast