



Tư duy tính toán



MIS



Giới thiệu môn học

Nội dung môn học

Môn học gồm một số chủ đề sau:

- Thuật toán và phân tích thuật toán.
- Một số thuật toán sắp xếp cơ bản.
- Một số thuật toán tìm kiếm cơ bản.
- Một số cấu trúc dữ liệu thường gặp (stack, queue, tree).
- Lý thuyết đồ thị.

Mục đích môn học

Có kiến thức cơ bản về thuật toán và cách phân tích thuật toán.

Có kiến thức về một số thuật toán (sắp xếp, tìm kiếm) cơ bản.

Có kiến thức về một số cấu trúc dữ liệu cơ bản.

Chấm điểm

- Điểm danh: 20%
- Thực hành: 30%
- Thi cuối kỳ: 50%.

Tài liệu tham khảo

<https://algs4.cs.princeton.edu/home/>

<https://realpython.com/python-data-structures/>

<https://freetuts.net/giai-thuat>

Thuật toán



Thuật toán là gì?

Thuật toán, còn gọi là giải thuật, là một tập hợp hữu hạn của các chỉ thị hay phương cách được định nghĩa rõ ràng cho việc hoàn tất một việc từ một trạng thái ban đầu cho trước; khi các chỉ thị này được áp dụng triệt để thì sẽ dẫn đến kết quả sau cùng như đã dự đoán trước.

Thuật toán là gì? (tiếp)

Nói cách khác, thuật toán là một bộ các quy tắc hay quy trình cụ thể nhằm giải quyết một vấn đề trong một số bước hữu hạn, hoặc nhằm cung cấp một kết quả từ một tập hợp của các dữ kiện đưa vào.

Ví dụ

```
#Cho thuật toán tìm số lớn nhất trong 2 số
def tìm_số_lớn_nhất(a, b)
    if a > b:
        return a
    else:
        return b
```

Tính chất của thuật toán

- **Tính chính xác:** để đảm bảo kết quả tính toán hay các thao tác mà máy tính thực hiện được là chính xác.
- **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.

Tính chất của thuật toán (tiếp)

- **Tính khách quan:** Một thuật toán dù được viết bởi nhiều người trên nhiều máy tính vẫn phải cho kết quả như nhau.
- **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.
- **Tính kết thúc:** Thuật toán phải gồm một số hữu hạn các bước tính toán.

Ví dụ

Như vậy, để tìm số lớn nhất trong hai số, ta thực hiện theo các bước:

1. So sánh hai số đó.
2. Nếu $a > b$ thì a là số lớn nhất trong 2 số.
3. Ngược lại thì b là số lớn nhất trong 2 số.

Phân loại thuật toán

Nếu không hiểu rõ thuật toán sẽ rất khó để phân loại chúng. Tùy thuộc vào hoàn cảnh sử dụng, các tiêu chí khác nhau mà thuật toán được phân ra nhiều loại.

Phân loại theo tính năng

- **Thuật toán tìm kiếm:** Đây là thuật toán được áp dụng để tìm kiếm dữ liệu, thông tin trong một tập hợp bao gồm các phần tử khác nhau.
- **Thuật toán sắp xếp:** Đây là thuật toán được dùng để sắp xếp thứ tự từng phần tử trong tập hợp một cách khoa học, đáp ứng yêu cầu ban đầu.
- **Thuật toán đồ thị:** Thuật này được sử dụng để xử lý các dạng bài có sử dụng đồ thị.

Phân loại theo cách thực hiện

- Chia để trị
- Quay lui
- Tham lam
- Đệ quy
- Quy hoạch động

Phân loại theo độ phức tạp

- Thời gian hằng số.
- Thời gian logarithm.
- Thời gian tuyến tính.
- Thời gian đa thức.
- Thời gian lũy thừa.

Cách biểu diễn thuật toán

Để có thể truyền đạt thuật toán cho người khác hay chuyển thuật toán thành chương trình máy tính, ta phải có phương pháp biểu diễn thuật toán. Có 3 phương pháp biểu diễn thuật toán:

- Dùng ngôn ngữ tự nhiên.
- Dùng lưu đồ - sơ đồ khối (flowchart)
- Dùng mã giả (pseudocode)

Ngôn ngữ tự nhiên

Trong cách biểu diễn thuật toán theo ngôn ngữ tự nhiên, người ta sử dụng ngôn ngữ thường ngày để liệt kê các bước của thuật toán. Phương pháp biểu diễn này không yêu cầu người viết thuật toán cũng như người đọc thuật toán phải nắm các quy tắc. Tuy vậy, cách biểu diễn này thường dài dòng, không thể hiện rõ cấu trúc của thuật toán, đôi lúc gây hiểu lầm hoặc khó hiểu cho người đọc. Gần như không có một quy tắc cố định nào trong việc thể hiện thuật toán bằng ngôn ngữ tự nhiên. Tuy vậy, để dễ đọc, ta nên viết các bước con lùì vào bên phải và đánh số bước theo quy tắc phân cấp như 1, 1.1, 1.1.1, ...

Ví dụ

Để tìm số lớn nhất trong ba số a , b và c . Ta dùng thuật toán sau:

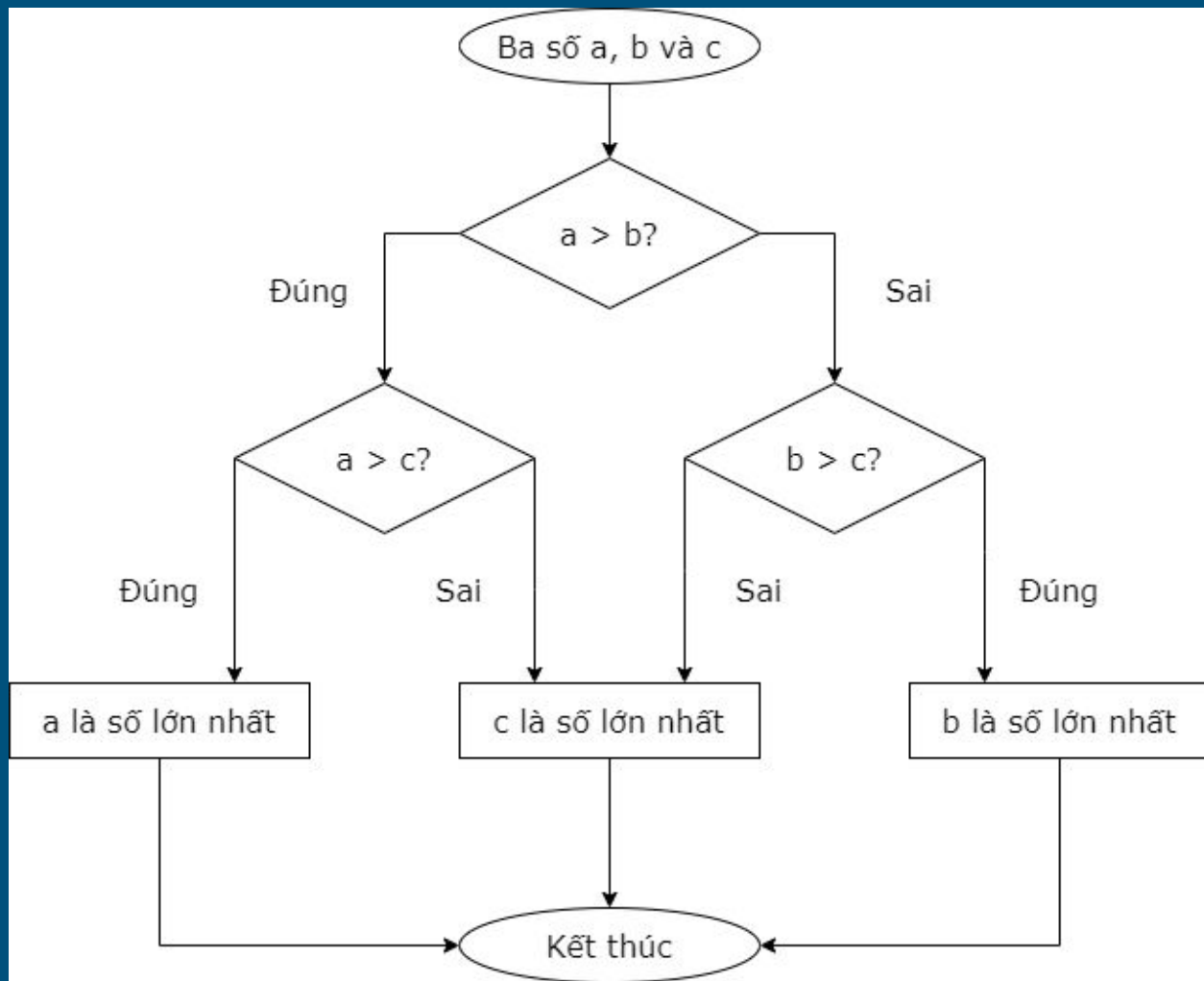
1. So sánh 2 số a và b để tìm ra số lớn hơn.
2. So sánh số lớn hơn đó với c để tìm ra số lớn nhất.

Sơ đồ khối

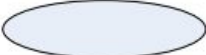
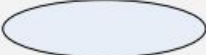







Lưu đồ hay sơ đồ khối là một công cụ trực quan để diễn đạt các thuật toán. Biểu diễn thuật toán bằng lưu đồ sẽ giúp người đọc theo dõi được sự phân cấp các trường hợp và quá trình xử lý của thuật toán. Phương pháp lưu đồ thường được dùng trong những thuật toán có tính rắc rối, khó theo dõi được quá trình xử lý.

Ví dụ

Chúng ta biểu diễn việc tìm số lớn nhất trong ba số bằng sơ đồ khối như sau:



Quy ước sơ đồ khối

STT	KÝ HIỆU	DIỄN GIẢI
1		Bắt đầu chương trình
2		Kết thúc chương trình
3		Luồng xử lý
4		Điều khiển lựa chọn
5		Nhập
6		Xuất
7		Xử lý, tính toán hoặc gán
8		Trả về giá trị (return)
9		Điểm nối liên kết tiếp theo (Sử dụng khi lưu đồ vượt quá trang)

Mã giả

Tuy sơ đồ khối thể hiện rõ quá trình xử lý và sự phân cấp các trường hợp của thuật toán nhưng lại cồng kềnh. Để mô tả một thuật toán nhỏ ta phải dùng một không gian rất lớn. Hơn nữa, lưu đồ chỉ phân biệt hai thao tác là rẽ nhánh (chọn lựa có điều kiện) và xử lý mà trong thực tế, các thuật toán còn có thêm các thao tác lặp.

Mã giả (tiếp)

Vì vậy, để biểu diễn một thuật toán phức tạp, người ta thường sử dụng mã giả (pseudocode).

Khi thể hiện thuật toán bằng mã giả, ta sẽ vay mượn các cú pháp của một ngôn ngữ lập trình nào đó để thể hiện thuật toán. Tất nhiên, mọi ngôn ngữ lập trình đều có những thao tác cơ bản là xử lý, rẽ nhánh và lặp. Dùng mã giả vừa tận dụng được các khái niệm trong ngôn ngữ lập trình, vừa giúp người cài đặt dễ dàng nắm bắt nội dung thuật toán.

Mã giả (tiếp)

Trong học phần này, chúng ta sẽ vay mượn cú pháp của Python để biểu diễn mã giả.

Ví dụ, để tìm số lớn nhất trong ba số a , b và c , chúng ta có thuật toán sau

Ví dụ

```
def tìm_số_lớn_nhất (a, b, c):  
    if a > b:  
        if a > c: return a  
        else: return c  
    else: #tức là b lớn hơn a  
        if b > c: return b  
        else: return c
```