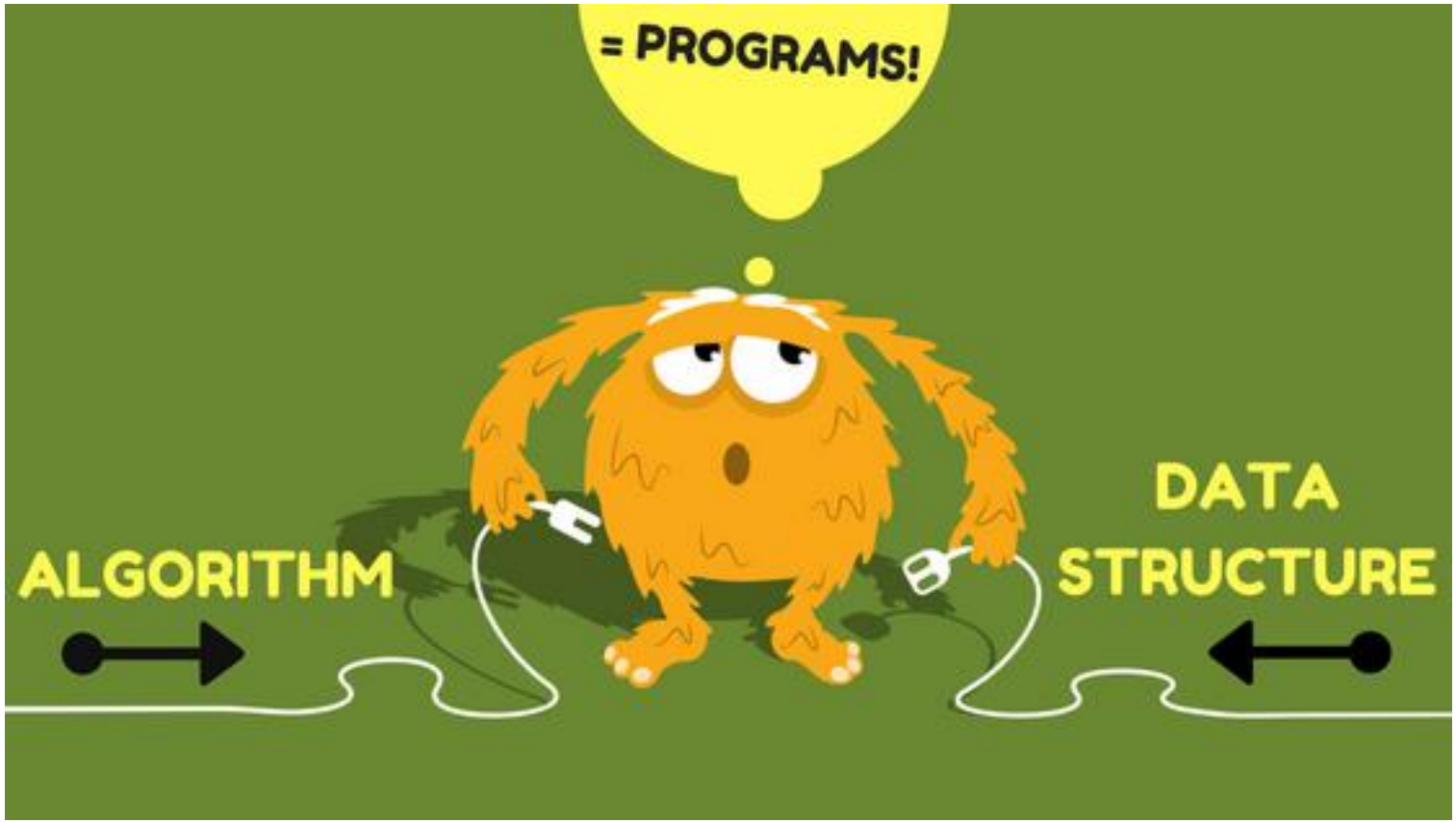
TOÁN RỜI RẠC VÀ THUẬT TOÁN

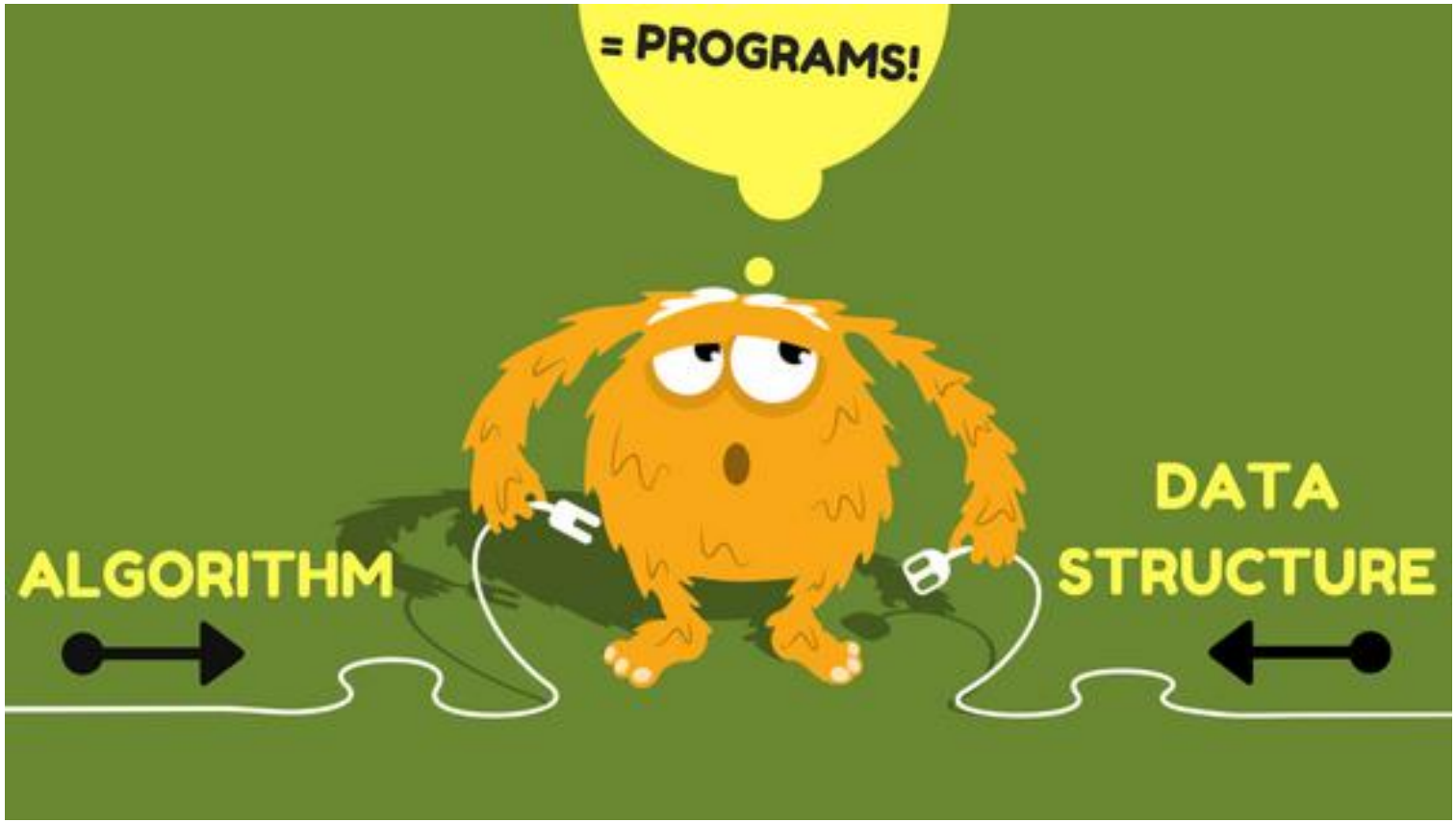
(DISCRETE MATHEMATIC AND ALGORITHMS)

Bài 1
Một số kiến thức cơ sở
(*Preliminaries*)

Nguyễn Thị Hồng Minh

minhnth@hus.edu.vn


= PROGRAMS!
ALGORITHM
DATA
STRUCTURE



"Thuật toán + Cấu trúc dữ liệu = Chương trình"
(Algorithms + Data Structures = Programs)

(Niklaus Wirth)

Nội dung

- **Một số khái niệm cơ bản**
- **Chương trình = Cấu trúc dữ liệu + Thuật toán**
- **Toán rời rạc và thuật toán**

Một số khái niệm cơ bản

- **Bài toán**

  Thuật ngữ "bài toán" trong tin học: lớp các bài toán cụ thể cùng loại

  - Ví dụ 1. Sắp xếp một dãy số theo thứ tự

    Cho: Một dãy số a= (a_1, a_2,..., a_n)

    Cần: Xác định dãy a' = (a'$_1$,a'$_2$,..,a'$_n$), a'$_i$ thuộc a, a'$_1$<=...<=a'$_n$
  - Ví dụ 2. Tìm đường đi ngắn nhất từ u tới v của đồ thị G=(V,E)

    Cho: Đồ thị G, đỉnh u,v

    Cần: xác định đường đi d = (u=v_1,v_2,...,v_n=v) (với v_i thuộc V (v_i,v_{i+1}) thuộc E) có độ dài ngắn nhất.

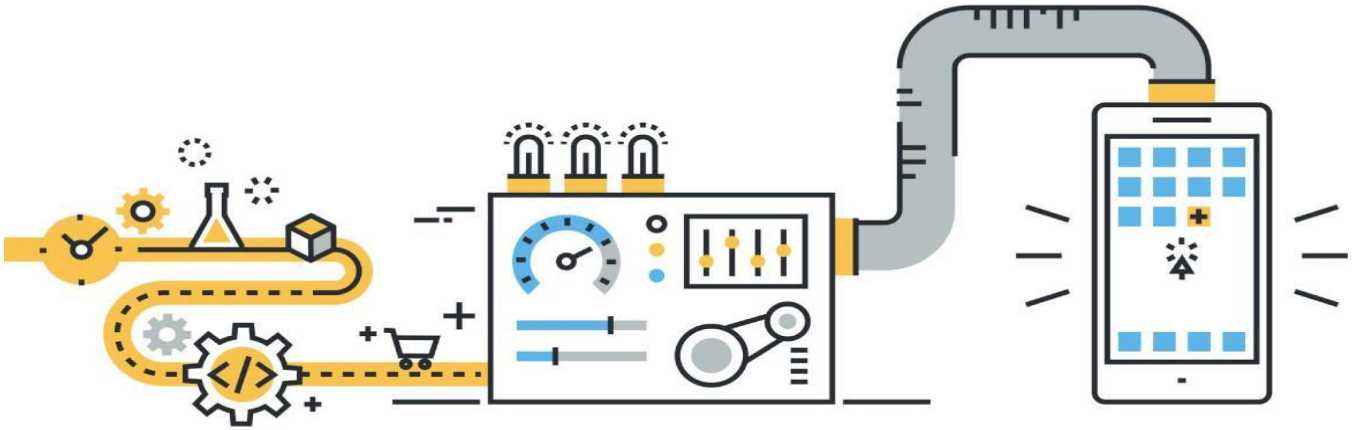
  - Bài toán được cấu tạo bởi hai thành phần cơ bản:
    - Thông tin vào (input): Cung cấp cho ta các dữ liệu đã có
    - Thông tin ra (output): Những yếu tố cần xác định.
  - ***Cho bài toán:*** cho input và ouput.
  - ***Giải bài toán:*** từ input, dùng một số hữu hạn các bước thao tác có cơ sở toán học thích hợp để tìm được output theo yêu cầu của bài toán đề ra.

Một số khái niệm cơ bản

- **Thuật toán**

  Thuật toán là một dãy hữu hạn các thao tác đơn giản được sắp xếp theo một trình tự xác định dùng để giải một bài toán.

  Dãy các thao tác đơn giản, có thể "giao cho máy tính làm được" để từ input dẫn ra output một cách tường minh



Một số khái niệm cơ bản

- **Dữ liệu**
  - Các đối tượng mà thuật toán sẽ sử dụng để đạt được kết quả mong muốn: input, ouput, dữ liệu trung gian (provisional data).
  - Hai đặc trưng của dữ liệu:
    - **Mặt tĩnh** (static): xác định kiểu dữ liệu (data type) – tổ chức dữ liệu, miền giá trị?
    - **Mặt động** (dynamic): là trạng thái của dữ liệu – tồn tại hay không, giá trị cụ thể?
  - ***Ví dụ:*** Dữ liệu cho bài toán tính tổng các phần tử của dãy số nguyên

Một số khái niệm cơ bản

- **Cấu trúc dữ liệu**
  - Kiểu dữ liệu: nhiều thành phần dữ liệu theo cấu trúc nào đó.
  - Biểu diễn cho các thông tin có cấu trúc của bài toán - khía cạnh logic của dữ liệu.
  - Phân biệt với: Dữ liệu vô hướng hay dữ liệu đơn giản
  - Cấu trúc dữ liệu ⇔ Thuật toán

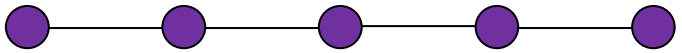
- **Chương trình**
  - Chương trình = thuật toán + cấu trúc dữ liệu.

Chương trình

- **Ngôn ngữ lập trình**
  - Cú pháp, ngữ nghĩa, môi trường (IDE)
  - Một số ngôn ngữ phổ biến
    - Pascal
    - C, C++
    - Java
    - Python
    - R, Mathlab…
- **Trao đổi chương trình**
  - Mã nguồn
  - Chương trình đã biên dịch: .exe, .dll

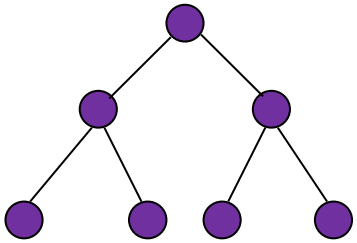
Cấu trúc dữ liệu

- **Hai loại cấu trúc dữ liệu**
  - CTDL tuyến tính:
    - Phần tử bố trí trật tự tuyến tính (có trước, sau)
    - Ví dụ: mảng (array), danh sách (list)

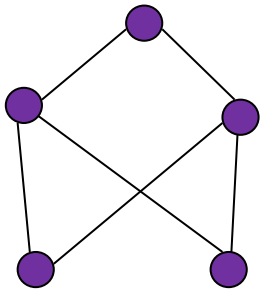


List

  - CTDL phi tuyến
    - Phần tử bố trí không theo thứ tự trước, sau
    - Ví dụ: tập hợp (set), cây (tree), đồ thị (graph)



Tree



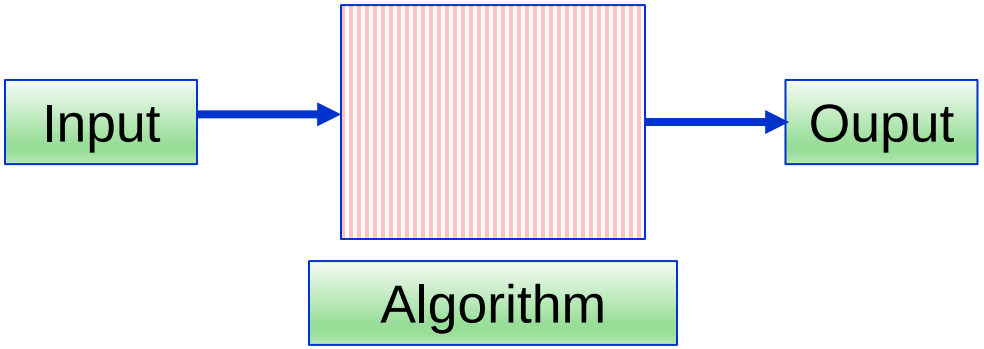
Graph

Cấu trúc dữ liệu

- **Cấu trúc lưu trữ (storage structure)**
  - Khái niệm: Khía cạnh vật lí (cài đặt) của một CTDL
    - Nguyên tắc: Cách tổ chức lưu trữ dữ liệu của máy tính
    - Thực tế: Cấu trúc kiểu dữ liệu mà ngôn ngữ lập trình hỗ trợ hoặc chương trình mô tả
    - Ví dụ: mảng (array), xâu kí tự (string), ngăn xếp (stack), hàng đợi (queue)
  - Hai cấu trúc lưu trữ chính
    - Cấu trúc lưu trữ trong: cài đặt trong bộ nhớ chính
    - Cấu trúc lưu trữ ngoài: cài đặt trên bộ nhớ phụ (ngoài)
    - Đặc trưng: Tốc độ truy cập; Kích thước; Tính lưu tồn
  - Cấu trúc lưu trữ trong
    - Tĩnh: Kích thước dữ liệu cố định, truy cập theo chỉ số
    - Động: Kích thước dữ liệu có thể thay đổi khi chạy CT, truy cập theo địa chỉ

Thuật toán

- **Thuật toán** – Các bước hữu hạn để giải bài toán



- **Các đặc trưng của thuật toán**
  - Tính tổng quát: áp dụng cho một lớp các bài toán
  - Tính dừng
  - Tính xác định
  - Tính hiệu quả

Thuật toán

- **Một số phương pháp diễn đạt thuật toán**
  - Liệt kê từng bước
  - Sơ đồ khối
  - Giả mã (pseudo-code)
- **Nguyên tắc diễn đạt thuật toán**
  - Tính logic: hiểu rõ tinh thần, các bước thuật toán
  - Tính có thể cài đặt: Khả năng có thể lập trình
  - Pseudo-code: Sử dụng ngôn ngữ tự nhiên tựa ngôn ngữ lập trình
- **Ví dụ**

B1. Lập danh sách các số dương; *B2. Sắp xếp DS theo thứ tự giảm dần;* *B3. Lấy ra phần tử đầu tiên của DS.*	*Input* ($a_1, a_2, \ldots a_n$); *Sort _decreasing*($a_1, a_2, \ldots a_n$); *Print* (a_1).

Các bước trên giải bài toán nào?

Thuật toán: Phân tích thuật toán

- **Phân tích để đánh giá thuật toán:**
  - Có đúng đắn không?
  - Có hiệu quả không?
- **Tính hiệu quả:**
  - Thời gian (số lượng các bước tính toán).

    ⇨ Độ phức tạp thời gian
  - Không gian (tài nguyên sẽ được sử dụng khi thuật toán thi hành).

    ⇨ Độ phức tạp không gian

☞ *Lecture 2*

Toán rời rạc và thuật toán

Toán rời rạc

- *"Định nghĩa":* Nhánh của toán học với đối tượng nghiên cứu, xử lí là các tập dữ liệu rời rạc
- Lịch sử:
  - Thế kỉ XVIII: bài toán tô màu đồ thị, mật mã...
  - Nửa sau thế kỉ XX, gắn với khoa học máy tính
  - Phản ánh và áp dụng thế giới thực và các bài toán nảy sinh từ thực tiễn
  - Giảng dạy đại học những năm 1980, phát triển thành lĩnh vực nghiên cứu rộng

Toán rời rạc và thuật toán

Toán rời rạc

- Mục tiêu nghiên cứu
  - Lí luận toán học (mathematical reasoning)
  - Phân tích kết hợp (combinatorial analysis)
  - Cấu trúc rời rạc (discrete structures): tập hợp, đồ thị, cây, otomat...
  - Tư duy thuật toán (algorithmic thinking)
  - Ứng dụng và mô hình hóa (applications and modeling)

Toán rời rạc và thuật toán

Toán rời rạc

- Topics
  - Logic mệnh đề (propositional logic)
  - Tập hợp (sets)
  - Lí thuyết số và mật mã (number theory and cryptography)
  - Đồ thị (graph)
  - Cây (tree)