

Cấu trúc dữ liệu và Giải thuật

(Data Structures and Algorithms)

Giảng viên: Tạ Việt Cường

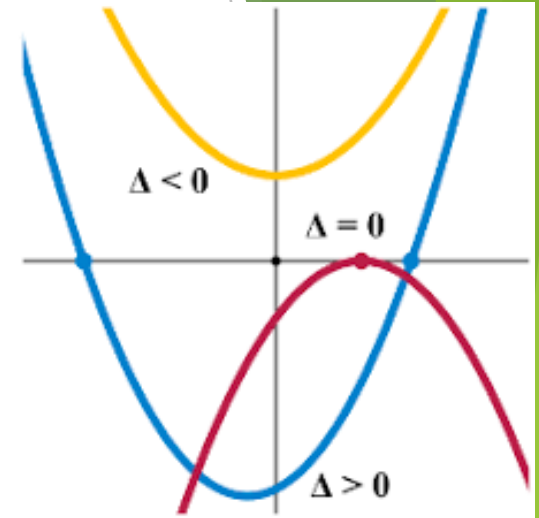
Phòng HMI - Khoa CNTT

Giới thiệu chung về môn DSA

- ▶ Cấu trúc dữ liệu: Data Structures
 - ▶ Cách lưu trữ thông tin trong máy tính
 - ▶ Ví dụ về 1 số loại thông tin:
 - ▶ Trường đhcn: thông tin về môn học, sinh viên, giảng viên...
 - ▶ Google Search: thông tin về url, nội dung website...
 - ▶ Facebook: thông tin danh sách bạn bè...
- ▶ Giải thuật: Algorithms – thuật toán
 - ▶ Phương pháp để giải 1 bài toán nhất định
 - ▶ **Input** - dữ liệu được đưa vào
 - ▶ **Output** - yêu cầu kết quả
 - ▶ Viết dưới dạng **pseudo-code** (mã giả)
 - ▶ Ví dụ: Giải phương trình bậc 2
 - ▶ Input: 3 số a, b, c thể hiện $ax^2 + bx + c = 0$
 - ▶ Output: tìm x

Ví dụ 1: $ax^2 + bx + c = 0$

- ▶ Bài toán giải phương trình bậc 2:
 - ▶ Dữ liệu: a, b, c
 - ▶ Cấu trúc dữ liệu: sử dụng 3 biến a, b, c
 - ▶ Hoặc: mảng có 3 phần tử $D[3] = [a, b, c]$
 - ▶ Giải thuật: viết dưới dạng pseudo-code
 - ▶ Step 1: Tính Δ (delta)
 - ▶ Step 2: So sánh Δ với 0
 - ▶ Step
- ▶ How about?
 - ▶ Phương trình bậc 3, bậc 4, bậc n



Pseudo-code(1)

- ▶ Cách thể hiện lời giải 1 bài toán:
 - ▶ Ngắn gọn, thường gồm các phép toán đơn giản:
 - ▶ cộng, trừ, nhân, chia
 - ▶ vòng lặp (lệnh for)
 - ▶ điều kiện (lệnh if)
 - ▶ Đủ chi tiết để có thể viết thành code
 - ▶ Không phụ thuộc vào ngôn ngữ lập trình
- ▶ Ví dụ 2: cho điểm số của 1 lớp học, tìm sv có điểm lớn nhất

Tên	Điểm DSA
Long	8.0
Đăng	7.0
Bình	9.0
Trang	9.5

Pseudo-code (2)

- ▶ Mô hình hóa bài toán với cấu trúc dữ liệu mảng
 - ▶ Input: **N** – số sinh viên trong lớp học
A[i] – điểm của sinh viên i
 - ▶ Output: Tìm vị trí của phần tử i, sao cho A[i] **lớn nhất**
- ▶ Pseudo-code:
 - ▶(**Easy**)

Tên	Điểm DSA
Long	8.0
Đăng	7.0
Bình	9.0
Trang	9.5

N=4
A=[8, 7, 9, 9.5]

i=3, A[i]=9.5

Tại sao phải học DSA

- ▶ Bài toán thực tế rất phức tạp:
 - ▶ Giải phương trình bậc N
 - ▶ Bài toán tìm thông tin của lớp, của trường:
 - ▶ Danh sách sinh viên, điểm số có thể thay đổi thường xuyên
 - ▶ Xử lý nhiều yêu cầu 1 lúc
- ▶ Một số ứng dụng khác:
 - ▶ Tìm kiếm 1 file trên máy tính
 - ▶ Tìm đường đi ngắn nhất/nhanh nhất từ Cầu Giấy lên Hồ Gươm



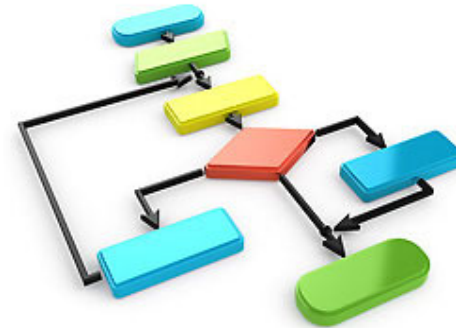
Nội dung môn học

► Khái niệm chung:

- Input/Output của 1 bài toán
- Viết hoặc làm theo 1 pseudo-code cụ thể
- Độ phức tạp thuật toán

► Cấu trúc dữ liệu (Data Structures):

- Mảng, Stack, Queue
- Tree
- Dictionary
- v..v..



► Giải thuật (Algorithms):

- Sắp xếp
- Tìm kiếm
- Đệ quy
- Các thuật toán trên đồ thị

Nội dung môn học

- ▶ Phương pháp học:

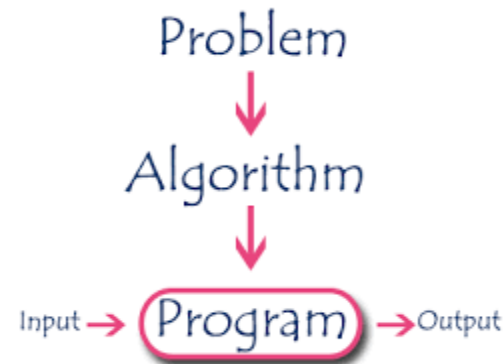
- ▶ Trừu tượng hơn Toán
 - ▶ Nhiều biến số với các điều kiện hơn
- ▶ Đọc hiểu (tốt nhất)
- ▶ Hoặc, học thuộc...

- ▶ Tài liệu tham khảo:

- ▶ Slide + code, <https://algs4.cs.princeton.edu/home/>
- ▶ Wikipedia

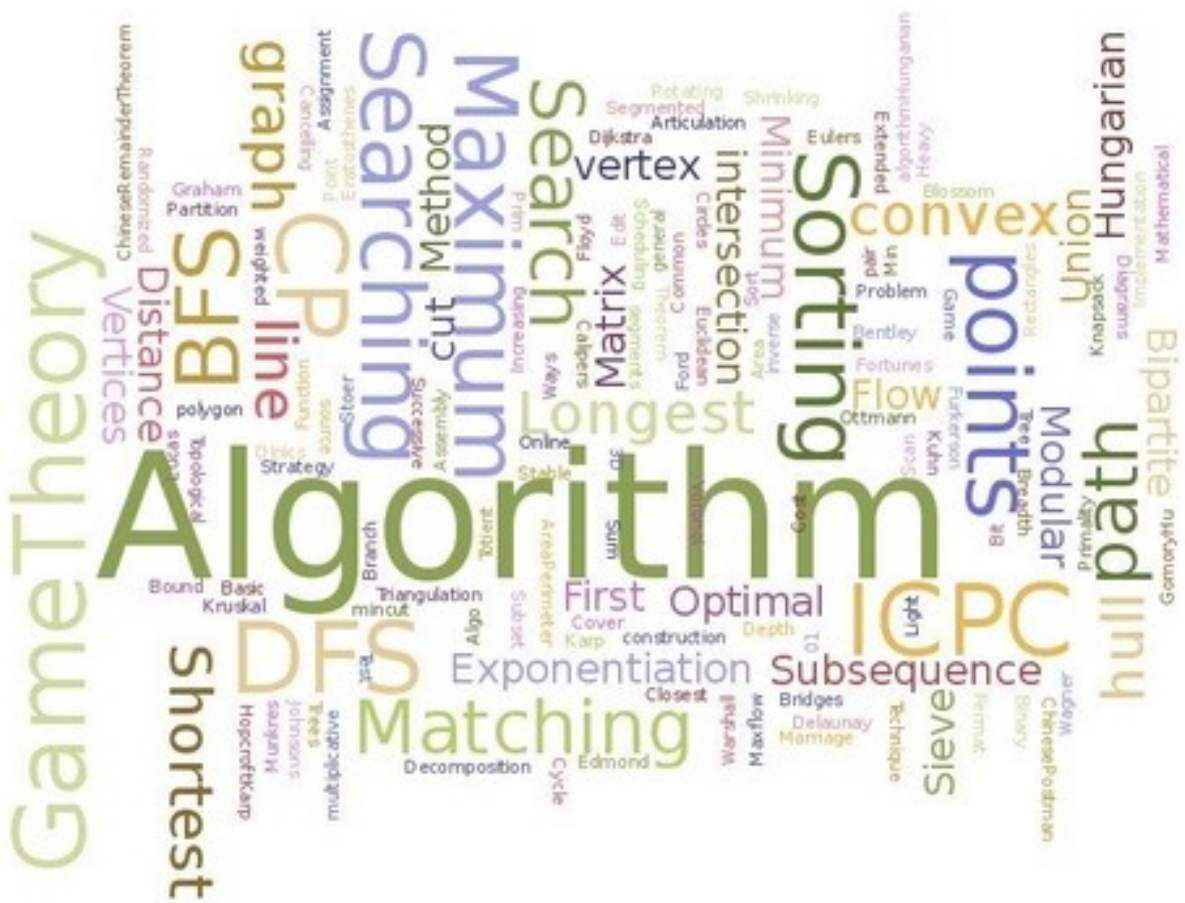
- ▶ Advanced resources:

- ▶ Google
- ▶ <https://www.hackerrank.com/>
- ▶ <https://codeforces.com/>



Nội dung môn học

- ▶ Thứ 5: lý thuyết + bài tập về nhà (maybe)
- ▶ Thứ 6: bài tập thực hành + thảo luận
 - ▶ Thực hành: bắt đầu vào 21/09/2018
- ▶ Điểm thành phần:
 - ▶ Điểm trên lớp: 20%
 - ▶ Điểm danh + gọi lên bảng chữa bài
 - ▶ Cần có mặt ít nhất 50% số buổi điểm danh
 - ▶ Điểm giữa kì: 20%
 - ▶ Trắc nghiệm hoặc tự luận
 - ▶ Điểm cuối kì: 60%
 - ▶ Đề thi chung



Which level?

❖ Beginner:

- ❖ Qua đc môn DSA (≥ 5 điểm)
- ❖ Hiểu đc: **stack, queue, tree**
- ❖ Học thuộc đc **6-8** pseudo-code



❖ Senior:

- ❖ Qua vòng phỏng vấn DSA ở **industry**
- ❖ Đọc hiểu **code/open source** về DSA
- ❖ **Đi dạy** DSA (?)



❖ Master:

- ❖ **Olympic tin học sinh viên**
- ❖ **ACM world final**
- ❖ **Google code jam onsite**

