**BÀI TẬP CHƯƠNG 1: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**Phần 1:** Câu hỏi:

**Câu 1:**

Trong Khoa học máy tính, một cấu trúc dữ liệu là hệ thống tổ chức sắp xếp dữ liệu trong đó việc lưu trữ và cách lưu trữ dữ liệu sao cho việc truy xuất và xử lí dữ liệu đó đạt hiệu quả cao nhất.

Ví dụ: Một cấu trúc cơ bản của một sinh viên (MSSV, Họ tên, Giới tính, Ngày sinh, Địa chỉ). Trong đó:

+ MSSV, Họ tên, Giới tính, Địa chỉ: có kiểu dữ liệu là kiểu chuỗi

+ Ngày sinh: có kiểu dữ liệu là kiểu Date (kiểu ngày)

**Câu 2:**

Trong Khoa học máy tính, thuật toán là một quy trình được định nghĩa và tính toán kỹ lưỡng, với đầu vào là giá trị nào đó hoặc tập các giá trị và trả ra kết quả cũng là giá trị nào đó hoặc tập các giá trị, hay còn gọi là đầu ra. Hay nói cách khác thuật toán chính là tập hợp tuần tự các bước tính toán, biến đổi đầu vào thành đầu ra.

Ví dụ: Thuật toán để giải phương trình bậc nhất y = ax + b;

Giải thuật của bài toán:

+ Đầu vào (Input): a, b (a, b R)

+ Đầu ra (Output): Kết luận nghiệm

Bước 1: Nhập hai số thực a, b

Bước 2: Thực hiện kiểm tra a

2.1 Nếu a = 0

2.1.1 Nếu b ≠ 0 thì phương trình vô nghiệm, rồi kết thúc;

2.1.2 Nếu b = 0 thì x = 0 rồi chuyển sang bước 4;

2.2 Nếu a ≠ 0

Thì phương trình có nghiệm x = -b / a;

**Câu 3:**

Cấu trúc dữ liệu và giải thuật có mối liên hệ chặt chẽ với nhau vì: cấu trúc dữ liệu và thuật toán kết hợp lại với nhau tạo nên một chương trình theo sơ đồ sau:

Cấu trúc dữ liệu + Giải thuật = Chương trình

Với một cấu trúc dữ liệu đã chọn, sẽ có những giải thuật tương ứng, phù hợp. Khi cấu trúc dữ liệu thay đổi thường giải thuật cũng phải thay đổi theo để tránh việc xử lý gượng ép, thiếu tự nhiên trên một cấu trúc không phù hợp. Hơn nữa, một cấu trúc dữ liệu tốt sẽ giúp giải thuật xử lý trên đó có thể phát huy tác dụng tốt hơn, vừa đáp ứng nhanh vừa tiết kiệm vật tư, giải thuật cũng dễ hiễu và đơn giản hơn.

**Câu 4:**

Lần 1 thực hiện so sánh 1 lần

Lần 2 thực hiện so sánh 4 lần

Lần 3 thực hiện so sánh 6 lần

…

Lần n - 2 thực hiện so sánh ((n - 2)2 / 2) + ((n - 2) / 2) lần

Lần n - 1 thực hiện so sánh ((n - 1)2 / 2) + ((n - 1) / 2) lần

Lần n thì thực hiện so sánh (n2 / 2) + (n / 2) lần

=>> O(n)