Bài tập chương 1

# Câu 1:

Cấu trúc dữ liệu là cấu trúc (sự tổ chức) của dữ liệu/thông tin lên trên máy tính, mà ở đó với cấu trúc này máy tính có thể xử lý được.

Cấu trúc này phải rõ ràng, xác định, các thành phần bên trong cấu trúc phải rõ ràng, và xác định.

Ví dụ: Cấu trúc cơ bản 1 file HTML bao gồm thẻ <head> và thẻ <body>

Trong đó:

Thẻ <Head> dùng để chứa những thẻ cung cấp thông tin cho trang web.

Thẻ <Body> dùng để định nghĩa phần thân tài liệu.

# Câu 2:

Trong khoa học máy tính giải thuật được hiểu: Giải thuật là một tập hợp hữu hạn của các bước (chỉ thị hay hành đọng) theo một trình tự, được xác định rõ ràng nhằm mục đích để giải quyết một bài toán nào đó (dựa vào những giá trị đầu vào gọi là “Input” và cho ra kết quả đầu ra là “Output”).

Ví dụ: Để so sánh 3 số a, b, c ta có giải thuật sau:

Input : a, b, c.

Output: Kết quả so sánh.

# Câu 3:

Ctdl và gt có quan hệ mật thiết với nhau vì:

* Ctdl là một cấu trúc, và các thành phần bên trong đó là các bước để thực hiện hay giải quyết 1 bài toán và các bước đó được xắp xếp theo 1 trình tự rõ ràng , xác định. Giải thuật là 1 tập hợp các chỉ thị hành động hay các bước theo 1 trình tự để giải quyết 1 bài toán. Qua đó ta có thể nói các thành phần bên trong ctdl là 1 giải thuật. Vì vậy ta có thể xác định rằng ctdl và gt có quan hệ mật thiết với nhau.
* Ví dụ: Để có thể tìm ra số lớn nhất giữa 3 số a, b và c thì ta phải thức hiện các bước sau : Bước 1: so sánh a và b, trả về kết quả là số lớn hơn. Trong ví dụ này ta sẽ giả định kết quả trả về là a.
* Bước 2:So sánh a và số còn lại trong trường hợp này là c. Nếu a > c thì kết quả trả về cuối cùng là a và ngược lại.
* Qua ví dụ trên ta có thể thấy bài toán phải được giải quyết theo 1 cấu trúc các bước gọi là ctdl. Đầu vào là a, b và c , và đầu ra là kết quả so sánh gọi là thuật giải.

# Câu 4:

Trong ví dụ 1.12 thì ta gọi a[j - 1] > x là phép toán cơ bản.

Với i = 1 thì j chạy 1 lần => a[j - 1] > x chạy 1 lần.

Với i = 2 thì j chạy 2 lần => a[j - 1] > xchạy 2 lần.

Với i = 3 thì j chạy 3 lần => a[j - 1] > x chạy 3 lần.

Tương tự với i = n – 1 thì j chạy (n - 1) lần => a[j - 1] > x chạy (n - 1) lần.

Suy ra: T(n) = (n - 1) + (n - 2) +…+ 3 + 2 + 1 = n2/2 + n/2 ≈ O(n2).