CÂU HỎI LÝ THUYẾT

Câu 1: Trong khoa học máy tính, cấu trúc dữ liệu được hiểu như thế nào? Cho ví dụ.

\_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 cũng phải rõ ràng và xác định.

Ví dụ: Cấu trúc dữ liệu của một mặt hàng là: mã số, tên sản phẩm, ngày nhập/ xuất kho, số lượng.

Trong đó:

\_ Mã số và tên sản phẩm kiểu chuỗi.

\_Ngày nhập/xuất kho kiểu date.

\_Số lượng kiểu int.

Câu 2: Trong khoa học máy tính, giải thuật được hiểu như thế nào? Cho ví dụ.

\_Giải thuật là một tập hữu hạn của các bước (chỉ thị hay hành động) 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 gọi là “ouput”).

Ví dụ: Tính tổng 2 số nguyên

Giải thuật (T):

\_Đầu vào (input): a, b (a, b € Z)

\_Đầu ra (output): tổng của a và b

Câu 3: Tại sao nói CTDL và GT có quan hệ mật thiết với nhau? Liệt kê 1 ví dụ nói về cách thiết kế cấu trúc dữ liệu sẽ ảnh hưởng đến giải thuật, giải thích tại sao?

\_Giải thuật phản ánh các phép xử lý , còn đối tượng xử lý của giải thuật lại là dữ liệu, chính dữ

liệu chứa đựng các thông tin cần thiết để thực hiện giải thuật. Để xác định được giải thuật phù

hợp cần phải biết nó tác động đến loại dữ liệu nào.

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.

Cấu trúc dữ liệu + Giải thuật = Chương trình

Ví dụ: Ta có 1 danh sách chứa thông tin của các sinh viên, mỗi sinh viên lại có nhiều thông tin

khác nhau

Ta có thể sử dụng mảng 1 chiều và mảng 2 chiều nhưng mảng 2 chiều có 1 cấu trúc lưu trữ hợp lí

và phù hợp hơn so với mảng 1 chiều. Vì vậy, giải thuật cũng tự nhiên và đơn giản hơn.

Câu 4: Đếm số phép so sánh trong giải thuật ở ví dụ 1.12.

j = 0 =>; 2 so sánh

j = 1 =>; 4 so sánh

j = 2 =>; 6 so sánh

…

j = k =>; 2k + 2 so sánh

n = 1 =>; 1 so sánh

n = 2 =>; 2 \* (2k + 2) so sánh

n = 3 =>; 3 \* (2k + 2) so sánh

…

n = k =>; k \* (2k + 2) so sánh

* Có k \* (2k + 2) phép so sánh trong giải thuật