Câu 1: Trong khoa học máy tính, cấu trúc dữ liệu là cấu trúc của dữ liệu/thông tin lên máy tính, mà ở đó với cấu trúc này máy tính có thể xử lý được.

-ví dụ: cấu trúc của một sinh viên: mssv(int), học và tên(chuỗi), ngày tháng năm sinh(daye), quên quán(chuỗi), địa chỉ(chuỗi)..

Câu 2: Trong khoa học máy tính, giải thuật là một tập hữu hạn theo một trình tự, được xác định rõ ràng nhằm mục đích đẻ giải quyết một bài toán nào đó ( input, output).

-ví dụ: tìm nghiệm của phương trình bậc 2: ax2 +bx +c=0

Input:giá trị biến a,b,c.

Output: kết luận nghiệm.

Bước 1: Tính delta= b2-4ac

Bước 2: Xét

Delta<0 : phương trình vô nghiệm

Delta=0: phương trình có nghiệm kép x1, x2=

Delta >0:phương trình có 2 nghiệm phân biệt

X1,2=

Câu 3: Cấu trúc dữ liệu và giải thuật có quan hệ mật thiết với nhau vì:

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.

Ví dụ:

Khi cài đặt các hàm số phức tạp trên máy tính. Nếu tính bằng cách khai triển chuỗi vô hạn thì độ chính xác cao hơn nhưng thời gian chậm hơn hàng tỉ lần so với phương pháp xấp xỉ. Trên thực tế việc tính toán luôn luôn cho phép chấp nhận một sai số nào đó nên các hàm số trong máy tính đều được tính bằng phương pháp xấp xỉ của giải tích số.

Xác định đúng yêu cầu bài toán là rất quan trọng bởi nó ảnh hưởng tới cách thức giải quyết và chất lượng của lời giải. Một bài toán thực tế thường cho bởi những thông tin khá mơ hồ và hình thức, ta phải phát biểu lại một cách chính xác và chặt chẽ để hiểu đúng bài toán.

Câu 4: Số phép so sánh trong ví dụ 1.12:

While:

J=0 có 2 lần so sánh

J=1 có 4 lần so sánh

...

J=k có 2k+2 lần so sánh

Vòng for:

n=1 có lần so sánh

n=1 có 1 lần so sánh

n=2 có 2 lần so sánh và 1 vòng lặp (2k+2)

....

n=k có k lần so sánh và (k-1) lần chạy vòng while (2k+2)

⬄ Tổng=k+(k-1)\*(2k+2)