Họ tên: Nguyễn Thành Ngọc

MSSV: 1851010082

Nhóm 7

**Bài Tập Chương 1**

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

VD: Mã số sinh viên, họ tên, giới tính, ngày sinh, địa chỉ (cấu trúc dữ liệu cơ bản của một sinh viên)

**Câu 2**: Trong khoa học máy tính, 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 “output”)

VD: Cho 3 số nguyên a, b ,c. Mô tả thuật giải số lớn nhất trong 3 số đã cho

Thuật giải (T):

Đầu vào: 3 số nguyên a, b, c

Đầu ra: Số lớn nhất trong 3 số

B1: Gán max = a;

B1: Nếu b > max thì gán max = b;

B3: Nếu c > max thì gán max = c;

Kết luận max

**Câu 3:**

Giải thuật = phép xử lí

Đối tượng của giải thuật chính là “dữ liệu” được tổ chức thành “các cấu trúc”

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 hiểu và đơn giản hơn.

VD: Quản lí danh bạ điện thoại

Dữ liệu gồm:

-Họ và tên

-Số điện thoại

Nếu danh bạ không có tổ chức gì cả thì dẫn đến giải thuật là tìm tuần tự từ đầu đến cuối

Nếu danh bạ (Họ và tên) tổ chức theo thứ tự a,b,c thì ta có thể tìm kiếm theo giải thuật tìm kiếm nhị phân (tìm kiểu từ điển: chia đôi và tìm trên nửa có chwuas kết quả)

-> Thời gian tìm kiếm sẽ nhanh hơn.

**Câu 4:**

Gọi a[j – 1] (\*) là phép toán cơ bản

Với i = 1, j chạy 1 lần => (\*) chạy 1 lần

Với i = 2, j chạy 2 lần => (\*) chạy 2 lần

Với i = 3, j chạy 3 lần => (\*) chạy 3 lần

….

Với i = n – 2, j chạy n – 2 lần => (\*) chạy n – 2 lần

Với i = n – 1, j chạy n – 1 lần => (\*) chạy n – 1 lần

T(n) = (n – 1) + (n – 2) + (n – 3) + … + 3 + 2 + 1 = n2/2 + n/2 ≈ O(n2).