**TRẢ LỜI CÂU HỎI – CHƯƠNG 1**

**Họ và tên:** Nguyễn Thị Yến Khương

**MSSV:** 1654050052 – **Nhóm: 04**

**Bài làm:**

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

Trong Khoa học máy tính, Cấu trúc dữ liệu là một hệ thống được tổ chức để sắp xếp dữ liệu/thông tin trên máy tính, mà ở đó việc lưu trữ với cấu trúc này sẽ giúp cho dữ liệu được truy xuất và xử lý một cách hiểu quả nhất. Một cấu trúc dữ liệu được thiết kế tốt cho phép thực hiện nhiều phép toán, sử dụng càng ít tài nguyên, thời gian xử lý và không gian bộ nhớ càng tốt. Ngoài ra, một cấu trúc dữ liệu và các thành phần bên trong cấu trúc dữ liệu đó phải luôn đảm bảo tính rõ ràng, xác định.

Thực tế cho thấy, việc khó khăn nhất khi triển khai một chương trình đó là chọn lựa cấu trúc dữ liệu. Vì lựa chọn này ảnh hưởng trực tiếp đến chất lượng và hiệu năng của kết quả cuối cùng của một chương trình, là yếu tố quyết định chương trình có được triển khai thành công và có hoạt động hiệu quả hay không.

Ví dụ Câu trúc dữ liệu: Cấu trúc dữ liệu cơ bản của một sinh viên gồm Mã số sinh viên, Họ và tên, Giới tính, Ngày sinh, Địa chỉ. Trong đó, Mã số sinh viên, Họ và tên, Giới tính, Địa chỉ có kiểu dữ liệu chuỗi (string), còn Ngày sinh có kiểu dữ liệu ngày (date).

**Câu 2: Trong Khoa học máy tính, Giải thuật được hiểu như thế nào? Cho ví dụ.**

Trong Khoa học máy tính, Giải thuật/Thuật giải/Thuật toán được định nghĩa là tập hữu hạn của các bước (chỉ thị hay hành động) theo trình tự xác định nhằm giải quyết một bài toán đã đặt ra. Một bài toán có các giá trị đầu vào (input) là giá trị hoặc tập các giá trị và cho kết quả đầu ra (output) cũng là một giá trị hoặc tập các giá trị tương ứng. Tóm lại, Giải thuật là một quy trình được định nghĩa và tính toán kỹ lưỡng để biết đổi đầu vào thành đầu ra.

Một thuật toán luôn tồn tại các tính chất như:

* *Tính đúng:* Giải thuật phải luôn cho ra kết quả đúng với các giá trị đầu vào tương ứng.
* *Tính dừng:* Đảm bảo cho không có trường hợp nào để Giải thuật lặp vô hạn.
* *Tính rõ ràng, xác định:* Các bước trong thuật toán phải được trình bày một cách rõ ràng, tường minh, không mập mờ, không ẩn bên trong các thao tác con.
* *Tính khách quan:* Giải thuật có thể được viết bằng nhiều ngôn ngữ khác nhau và từ nhiều người khác nhau, nhưng phải luôn cho ra kết quả đúng và giống nhau.

Ví dụ về Giải thuật: Cho phương trình có dạng *ax + b = 0 (a ≠ 0)* được gọi là phương trình bậc nhất một ẩn.

Giải thuật (T) có:

* Đầu vào (input): a, b (với a ≠ 0)
* Đầu ra (output): x
* Bước 1: Chuyển vế *ax = −b*
* Bước 2: Chia hai vế cho *a*, ta được: *x = (−b)/a*
* Bước 3: Kết luận nghiệm: *S = {(−b)/a}*

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

Trong thực tế, các thành phần dữ liệu rất đa dạng, phong phú và thường có mối quan hệ mật thiết với nhau. Do đó, với mỗi bài toán được đặt ra, cần phải có cách tổ chức, xây dựng cấu trúc thích hợp cho vừa có thể phản ánh chính xác dữ liệu thực tế, vừa có thể dễ dàng xử lý trên máy tính. Công việc này được gọi là xây dựng Cấu trúc dữ liệu cho bài toán. Từ những yêu cầu thực tế trên, ta cần tìm ra các giải thuật tương ứng để xác định trình tự các thao tác mà máy tính phải thực thi để có thể cho ra kết quả mong muốn. Đây là bước xây dựng Giải thuật cho bài toán.

Tuy nhiên, khi giải quyết một bài toán trên máy tính, chúng ta thường có khuynh hướng chỉ chú trọng đến việc xây dựng Giải thuật mà quên đi tầm quan trọng của việc tổ chức Cấu trúc dữ liệu cho bài toán. 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, thì cần phải biết nó tác động đến loại dữ liệu nào, và khi chọn lựa Cấu trúc dữ liệu thì cũng cần phải hiểu rõ những thao tác nào sẽ tác động đến chính nó. Như vậy, trong một bài toán, Cấu trúc dữ liệu và Giải thuật có mối quan hệ mật thiết và chặt chẽ với nhau, và được thể hiện qua công thức:

**CẤU TRÚC DỮ LIỆU + GIẢI THUẬT = CHƯƠNG TRÌNH**

Với một Cấu trúc dữ liệu đã chọn, sẽ có những Giải thuật tương ứng sao cho phù hợp nhất. Khi Cấu trúc dữ liệu thay đổi, thường Giải thuật cũng phải thay đổi theo để tránh việc 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 tài nguyên của máy tính, lại còn dễ hiểu và đơn giản hơn. Tóm lại, cấu trúc dữ liệu nào thì giải thuật đó. Khi cấu trúc dữ liệu thay đổi, giải thuật cũng thay đổi theo.

Ví dụ chứng minh Cấu trúc dữ liệu và Giải thuật có mối quan hệ với nhau:

Trong Danh sách đặc, người ta thường sử dụng vòng lặp For để truy xuất và thao tác trên dữ liệu, bởi đặc điểm của Danh sách đặc là các phần tử được cấp phát liên tục trong bộ nhớ và ta có thể biết trước được thứ tự của các phần tử. Vậy nên, đối với cấu trúc dữ liệu Danh sách đặc, vòng lặp For sẽ là giải thuật lý tưởng để thực hiện các thao tác trên nó.

Còn đối với Danh sách liên kết, các phần tử được cấp phát rời rạc nhau và cố định trên bộ nhớ. Thế nên trong trường hợp này, vòng lặp For sẽ không còn phù hợp với cấu trúc dữ liệu này, mà thay vào đó, để thao tác và truy xuất dữ liệu trên danh sách liên kết, người ta thường sử dụng vòng lặp While.

Nếu ta cố ý sử dụng vòng lặp For cho cấu trúc dữ liệu Danh sách liên kết, và vòng lặp While cho cấu trúc dữ liệu Danh sách đặc thì sẽ gây ra sự gượng ép, thiếu tự nhiên và kém hiệu quả trong việc thao tác và xử lý dữ liệu.

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

* Đếm phép so sánh ở vòng lặp While:
* Xét j = 1: có 3 phép so sánh
* Xét j = 2: có 5 phép so sánh
* Xét j = 3: có 7 phép so sánh
* Xét j = k: có (2\*k + 1) phép so sánh
* Đếm phép so sánh ở vòng lặp For:
* Xét n = 0: có 1 phép so sánh và 0 vòng lặp While
* Xét n = 1: có 2 phép so sánh và 1 vòng lặp While
* Xét n = 2: có 3 phép so sánh và 2 vòng lặp While
* Xét n = k: có (k + 1) phép so sánh và k vòng lặp While

Vậy, tổng số phép so sánh: T = (2\*k + 1)\*k + (k + 1) ≈ n2 → Độ phức tập của thuật toán thuộc lớp O(n2).