**Chương 1**

Câu 1:

Trong khoa học máy tính, cấu trúc dữ liệu là một cách lưu dữ liệu trong máy tính sao cho máy tính có thể xử lý 1 cách hiệu quả.

Ví dụ: Cấu trúc dữ liệu cơ bản của 1 thư viện (mã sách, tên sách, tên tác giả, ngày mượn, ngày trả)

Trong đó:

* Mã sách, tên sách, tên tác giả có kiểu dữ liệu là kiểu chuỗi.
* Ngày mượn, ngày trả có kiểu dữ liệu là kiểu Date (kiểu ngày).

Câu 2:

Trong khoa học máy tính, giải thuật là tập hợp các bước để giải một bài toán dựa vào giá trị đưa vào (input) để cho ra kết quả (output).

Ví dụ: Bài toán sắp xếp dãy số theo thứ tự giảm dần

Input: Dãy số bất kỳ gồm n phần tử (a[n])

Output: Dãy số đã sắp xếp giảm dần

Bước 1: i = 0

Bước 2: j = n – 1

+ Lặp lại trong khi (j > i)

Nếu a[j] > a[j-1]

Đổi chỗ a[j] và a[j-1]

Bước 3:i++;

Nếu i< n-1 thì lặp lại bước 2.

Câu 3:

Trong 1 bài toán, giải thuật phản ánh các phép xử lý, đối tượng xử lý của giải thuật là dữ liệu. Để xác định được giải thuật phù hợp phải biết nó tác động đến loại dữ liệu nào và khi lựa chọn cấu trúc dữu liệu cũng cần phải hiểu rõ nhưng thao tác nào tác động đến nó. Như vậy trong 1 bài toán, cấu trúc dữ liệu và giải thuật có mối quan hệ chặt chẽ với nhau.

Ví dụ: Bài toán nhập điểm số của các sinh viên (Điểm toán, lý, hóa):

* Cách 1: Khai báo danh sách điểm của sinh viên bằng mảng 1 chiều ta lưu trữ dữ liệu điểm của sinh viên theo dạng:

9 8 6 9 5 7 4 8 10

**Sinh viên 3**

**Sinh viên 2**

**Sinh viên 1**

* Cách 2: Khai báo danh sách điểm của sinh viên bằng mảng 2 chiều ta lưu trữ dữ liệu điểm của sinh viên theo dạng:

|  |  |  |  |
| --- | --- | --- | --- |
| Sinh viên | Toán | Lý | Hóa |
| 1 | 9 | 8 | 6 |
| 2 | 9 | 5 | 7 |
| 3 | 4 | 8 | 10 |

* Giải thuật của cách 1 có độ phức tạp là O(n), giải thuật cách 2 có độ phức tạp là O(n2). Vì vậy cách thiết kế cấu trúc dữ liệu sẽ ảnh hưởng đến giải thuật.

Câu 4: Đếm số phép so sánh:

Vòng while:

* Với j = 0: có 1 phép so sánh
* Với j = 1: có 3 phép so sánh
* Với j = 2: có 5 phép so sánh
* Với j = 3: có 7 phép so sánh
* Với j = k: có (2k + 1) phép so sánh

T(n) = (2n + 1)

Vòng for:

* Với n = 0: có 1 phép so sánh
* Với n = 1: có 2 phép so sánh
* Với n = 2: có 3 phép so sánh
* Với n = k: có (k + 1) phép so sánh

T(n) = (n + 1)