**CHƯƠNG 1 BÀI TẬP LÝ THUYẾT**

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

Trong khoa học máy tính, một cấu trúc dữ liệu là 1 hệ thống tổ chức sắp xếp dữ liệu trong đó việc luư trữ và cách lưu trữ dữ liệu sao cho việc truy xuất và xử lí dữ liệu đó đạt hiệu quả cao nhất.

VD: Sử dụng cấu trúc dữ liệu để tạo danh sách công nhân, sinh viên……

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

Là một quy trình được định nghĩa và tính toán kỹ lưỡng,với đầu vào là giá trị nào đó hoặc tập các giá trị, và trả ra KQ cũng là giá trị nào đó hay tập các giá trị, hay còn được gọi là đầu ra. Hay nói cách khác, thuật toán là tập hợp tuần tự các bước tính toán, biến đổi đầu vào và thành đầu ra.

**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?**

Từ những yêu cầu xử lý thực tế, cần tìm ra các giải thuật tương ứng để xác định trình tự các thao tác máy tính phải thi hành để 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 dữ liệu trong 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 cần phải biết nó tác động đến loại dữ liệu nào (ví dụ để làm nhuyễn các hạt đậu , người ta dùng cách xay chứ không băm bằng dao, vì đậu sẽ văng ra ngoài) và khi chọn lựa cấu trúc dữ liệu cũng cần phải hiểu rõ những thao tác nào sẽ tác động đến nó (ví dụ để biểu diễn các điểm số của sinh viên người ta dùng số thực thay vì chuỗi ký tự vì còn phải thực hiện thao tác tính trung bình từ những điểm số đó). Như vậy trong một đề án tin học, giải thuật và cấu trúc dữ liệu có mối quan hệ chặt chẽ với nhau, 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, 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 1: Một chương trình quản lý điểm thi của sinh viên cần lưu trữ các điểm số của 3 sinh viên. Do mỗi sinh viên có 4 điểm số ứng với 4 môn học khác nhau nên dữ liệu có dạng bảng như sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sinh viên | Môn 1 | Môn 2 | Môn3 | Môn4 |
| SV 1 | 7 | 9 | 5 | 2 |
| SV 2 | 5 | 0 | 9 | 4 |
| SV 3 | 6 | 3 | 7 | 4 |

Chỉ xét thao tác xử lý là xuất điểm số các môn của từng sinh viên.

Giả sử có các phương án tổ chức lưu trữ sau:

Phương án 1 : Sử dụng mảng một chiều

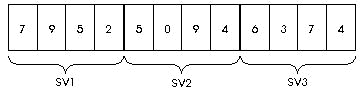
Có tất cả 3(SV)\*4(Môn) = 12 điểm số cần lưu trữ, do đó khai báo mảng *result* như sau :

int result [ 12 ] = {7, 9, 5, 2,

5, 0, 9, 4,

6, 3, 7, 4};

khi đó trong mảng *result* các phần tử sẽ được lưu trữ như sau:



Và truy xuất điểm số môn j của sinh viên i - là phần tử tại (dòng i, cột j) trong bảng - phải sử dụng một công thức xác định chỉ số tương ứng trong mảng result:

bảngđiểm(dòng i, cột j) Þ result[((i-1)\*số cột) + j]

Ngược lại, với một phần tử bất kỳ trong mảng, muốn biết đó là điểm số của sinh viên nào, môn gì, phải dùng công thức xác định sau

result[ i ] Þ bảngđiểm (dòng((i / số cột) +1), cột (i % số cột) )

Phương án 2 : Sử dụng mảng 2 chiều

Khai báo mảng 2 chiều *result* có kích thước 3 dòng\* 4 cột như sau :

int result[3][4] ={{ 7, 9, 5, 2},

{ 5, 0, 9, 4},

{ 6, 3, 7, 4 }};

khi đó trong mảng *result* các phần tử sẽ được lưu trữ như sau :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cột 0 | Cột 1 | Cột 2 | Cột 3 |
| Dòng 0 | result[0][0] =7 | result[0][1] =9 | result[0][2] =5 | result[0][3] =2 |
| Dòng 1 | result[1][0] =5 | result[1][1] =0 | result[1][2] =9 | result[1][3] =4 |
| Dòng 2 | result[2][0] =6 | result[2][1] =3 | result[2][2] =7 | result[2][3] =4 |

Và truy xuất điểm số môn j của sinh viên i - là phần tử tại (dòng i, cột j) trong bảng - cũng chính là phần tử nằm ở vị trí (dòng i, cột j) trong mảng

bảngđiểm(dòng i,cột j) Þ result[ i] [j]

Có thể thấy rõ phương án 2 cung cấp một cấu trúc lưu trữ phù hợp với dữ liệu thực tế hơn phương án 1, và do vậy giải thuật xử lý trên cấu trúc dữ liệu của phương án 2 cũng đơn giản, tự nhiên hơn.

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

\*Trong vòng lặp while:

* j=0 -> 2 lần ss
* j=1 -> 4 lần ss
* j=2 -> 6 lần ss
* j=k -> 2k+2 lần ss

Trong vòng lặp for:

* n=0 -> 1 lần ss
* n=1 -> 1 lần ss
* n=2 -> 2 lần ss và 1 lần chạy vòng lặp while \*(2k+2)
* n=k -> k lần ss và (k-1) lần chạy vòng lặp while \*(2k+2)

=> Tổng phép ss = k+(k-1)\*(2k+2) = 2k2+k-2