**8.1. BÀI TOÁN SẮP XẾP**

Sắp xếp là quá trình bố trí lại các phần tử của một tập đối tượng nào đó theo một thứ tự nhất định. Chẳng hạn như thứ tự tăng dần (hay giảm dần) đối với một dãy số, thứ tự từ điển đối với các từ v.v… Yêu cầu về sắp xếp thường xuyên xuất hiện trong các ứng dụng Tin học với các mục đích khác nhau: sắp xếp dữ liệu trong máy tính để tìm kiếm cho thuận lợi, sắp xếp các kết quả xử lý để in ra trên bảng biểu v.v…

Nói chung, dữ liệu có thể xuất hiện dưới nhiều dạng khác nhau, nhưng ở đây ta quy ước: Một tập các đối tượng cần sắp xếp là tập các bản ghi (records), mỗi bản ghi bao gồm một số trường (fields) khác nhau. Nhưng không phải toàn bộ các trường dữ liệu trong bản ghi đều được xem xét đến trong quá trình sắp xếp mà chỉ là một trường nào đó (hay một vài trường nào đó) được chú ý tới thôi. Trường như vậy ta gọi là khoá (key). Sắp xếp sẽ được tiến hành dựa vào giá trị của khoá này.

*Ví dụ: Hồ sơ tuyển sinh của một trường Đại học là một danh sách thí sinh, mỗi thí sinh có tên, số báo danh, điểm thi. Khi muốn liệt kê danh sách những thí sinh trúng tuyển tức là phải sắp xếp các thí sinh theo thứ tự từ điểm cao nhất tới điểm thấp nhất. Ở đây khoá sắp xếp chính là điểm thi.*

|  |  |  |  |
| --- | --- | --- | --- |
| *STT* | *SBD* | *Họ và tên* | *Điểm thi* |
| *1* | *A100* | *Nguyễn Văn A* | *20* |
| *2* | *B200* | *Trần Thị B* | *25* |
| *3* | *X150* | *Phạm Văn C* | *18* |
| *4* | *G180* | *Đỗ Thị D* | *21* |

Khi sắp xếp, các bản ghi trong bảng sẽ được đặt lại vào các vị trí sao cho giá trị khoá tương ứng với chúng có đúng thứ tự đã ấn định. Vì kích thước của toàn bản ghi có thể rất lớn, nên nếu việc sắp xếp thực hiện trực tiếp trên các bản ghi sẽ đòi hỏi sự chuyển đổi vị trí của các bản ghi, kéo theo việc thường xuyên phải di chuyển, copy những vùng nhớ lớn, gây ra những tổn phí thời gian khá nhiều. Thường người ta khắc phục tình trạng này bằng cách xây dựng một bảng khoá: Mỗi bản ghi trong bảng ban đầu sẽ tương ứng với một bản ghi trong **bảng khoá**. Bảng khoá cũng gồm các bản ghi nhưng mỗi bản ghi chỉ gồm có hai trường:

Trường thứ nhất chứa khoá

Trường thứ hai chứa liên kết tới một bản ghi trong bảng ban đầu, tức là chứa một thông tin đủ để biết bản ghi tương ứng với nó trong bảng ban đầu là bản ghi nào.

Sau đó, việc sắp xếp được thực hiện trực tiếp trên bảng khoá, trong quá trình sắp xếp, **bảng chính không hề bị ảnh hưởng gì**, việc truy cập vào một bản ghi nào đó của bảng chính vẫn có thể thực hiện được bằng cách dựa vào trường liên kết của bản ghi tương ứng thuộc bảng khoá.

*Như ở ví dụ trên, ta có thể xây dựng bảng khoá gồm 2 trường, trường khoá chứa điểm và trường liên kết chứa số thứ tự của người có điểm tương ứng trong bảng ban đầu:*

|  |  |
| --- | --- |
| *Điểm thi* | *STT* |
| *20* | *1* |
| *25* | *2* |
| *18* | *3* |
| *21* | *4* |

*Sau khi sắp xếp theo trật tự điểm cao nhất tới điểm thấp nhất, bảng khoá sẽ trở thành:*

|  |  |
| --- | --- |
| *Điểm thi* | *STT* |
| *25* | *2* |
| *20* | *1* |
| *21* | *4* |
| *18* | *3* |

*Dựa vào bảng khoá, ta có thể biết được rằng người có điểm cao nhất là người mang số thứ tự 2, tiếp theo là người mang số thứ tự 4, tiếp nữa là người mang số thứ tự 1, và cuối cùng là người mang số thứ tự 3, còn muốn liệt kê danh sách đầy đủ thì ta chỉ việc đối chiếu với bảng ban đầu và liệt kê theo thứ tự 2, 4, 1, 3.*

Có thể còn cải tiến tốt hơn dựa vào nhận xét sau: Trong bảng khoá, nội dung của trường khoá hoàn toàn có thể suy ra được từ trường liên kết bằng cách: Dựa vào trường liên kết, tìm tới bản ghi tương ứng trong bảng chính rồi truy xuất trường khoá trong bảng chính. Như ví dụ trên thì người mang số thứ tự 1 chắc chắn sẽ phải có điểm thi là 20, còn người mang số thứ tự 3 thì chắc chắn phải có điểm thi là 18. Vậy thì bảng khoá có thể loại bỏ đi trường khoá mà chỉ giữ lại trường liên kết. Trong trường hợp các phần tử trong bảng ban đầu được đánh số từ 1 tới n và trường liên kết chính là số thứ tự của bản ghi trong bảng ban đầu như ở ví dụ trên, người ta gọi kỹ thuật này là kỹ thuật **sắp xếp bằng chỉ số**: Bảng ban đầu không hề bị ảnh hưởng gì cả, việc sắp xếp chỉ đơn thuần là đánh lại chỉ số cho các bản ghi theo thứ tự sắp xếp. Cụ thể hơn:

Nếu r[1], r[2], …, r[n] là các bản ghi cần sắp xếp theo một thứ tự nhất định thì việc sắp xếp bằng chỉ số tức là xây dựng một dãy Index[1], Index[2], …, Index[n] mà ở đây:

Index[j] = Chỉ số của bản ghi sẽ đứng thứ j khi sắp thứ tự

(Bản ghi r[index[j]] sẽ phải đứng sau j - 1 bản ghi khác khi sắp xếp)

Do khoá có vai trò đặc biệt như vậy nên sau này, khi trình bày các giải thuật, ta sẽ coi **khoá như đại diện cho các bản ghi** và để cho đơn giản, ta chỉ nói tới giá trị của khoá mà thôi. Các thao tác trong kỹ thuật sắp xếp lẽ ra là tác động lên toàn bản ghi giờ đây chỉ làm trên khoá.

Còn việc cài đặt các phương pháp sắp xếp trên danh sách các bản ghi và kỹ thuật sắp xếp bằng chỉ số, ta coi như bài tập.

**Bài toán sắp xếp giờ đây có thể phát biểu như sau:**

Xét quan hệ thứ tự toàn phần "nhỏ hơn hoặc bằng" ký hiệu "" trên một tập hợp S, là quan hệ hai ngôi thoả mãn bốn tính chất:

Với a, b, c  S

Tính phổ biến: Hoặc là a  b, hoặc b  a;

Tính phản xạ: a  a

Tính phản đối xứng: Nếu a  b và b  a thì bắt buộc a = b. Tính bắc cầu: Nếu có a  b và b  c thì a  c.

Trong trường hợp a  b và a  b, ta dùng ký hiệu "<" cho gọn

Cho một dãy gồm n khoá. Giữa hai khoá bất kỳ có quan hệ thứ tự toàn phần "". Xếp lại dãy các khoá đó để được dãy khoá thoả mãn k1 k2  … kn.

Giả sử cấu trúc dữ liệu cho dãy khoá được mô tả như sau:

const

n = …; {Số khoá trong dãy khoá, có thể khai dưới dạng biến số nguyên để tuỳ biến hơn}

type

TKey = …; {Kiểu dữ liệu một khoá}

TArray = array[1..n] of TKey; var

k: TArray; {Dãy khoá}

Thì những thuật toán sắp xếp dưới đây được viết dưới dạng thủ tục sắp xếp dãy khoá k, kiểu chỉ số đánh cho từng khoá trong dãy có thể coi là số nguyên Integer.