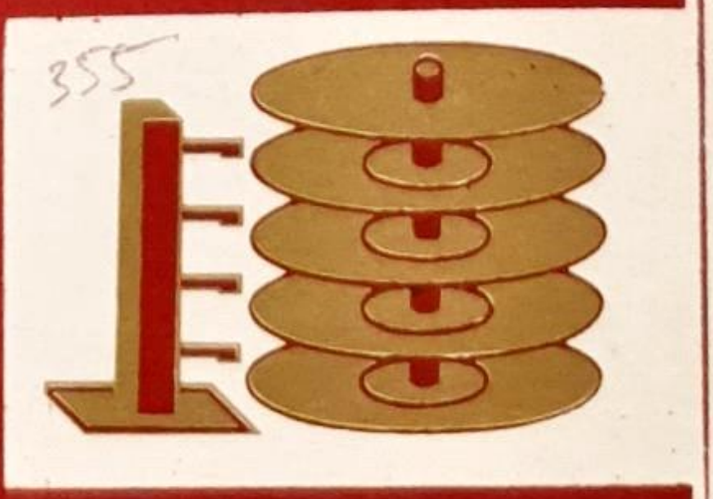
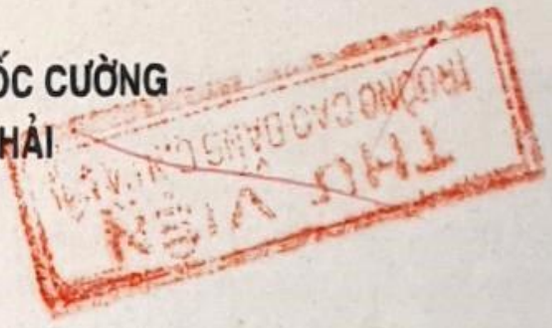
NGUYỄN QUỐC CƯỜNG
HOÀNG ĐỨC HẢI

CẤU TRÚC DỮ LIỆU + GIẢI THUẬT = CHƯƠNG TRÌNH



NHÀ XUẤT BẢN GIÁO DỤC

NGUYỄN QUỐC CƯỜNG
HOÀNG ĐỨC HẢI



CẤU TRÚC DỮ LIỆU + GIẢI THUẬT = CHƯƠNG TRÌNH

(Tái bản lần thứ 4)

NHÀ XUẤT BẢN GIÁO DỤC – 1999

LỜI TỰA

Lập trình cho máy tính đã được công nhận là bộ môn khoa học mà việc nắm vững nó là cơ sở và quyết định cho sự thành công của nhiều công trình. E. W. Dijkstra và C A. R. Hoare đã có nhiều đóng góp đáng kể. Bài "Một số điểm về lập trình cấu trúc" đã mở ra quan điểm mới về lập trình như là một môn khoa học, và đã tạo ra cuộc "cách mạng" trong lập trình. Bài "tiên đề cơ sở cho lập trình cấu trúc" của Hoare đã cho thấy rằng chương trình có thể tuân theo sự phân tích chính xác theo suy diễn toán học. Cả hai bài báo đã thuyết phục người đọc được rằng nhiều lỗi lập trình có thể tránh được nếu giúp cho người lập trình biết được các phương pháp và kĩ thuật mà trước giờ họ thường áp dụng không có ý thức. Các bài báo này tập trung vào khía cạnh phân tích và xây dựng chương trình hay chính xác hơn là cấu trúc của thuật giải được thể hiện bằng văn bản chương trình. Hơn nữa, việc tiếp cận có hệ thống và khoa học để xây dựng chương trình có tác dụng thật lớn khi các chương trình phức tạp và dữ liệu cũng phức tạp. Do đó một phương pháp lập trình cũng phải xét tới mọi khía cạnh của cấu trúc dữ liệu. *Chương trình*, chính là mô tả cụ thể của các *thuật giải* trừu tượng dựa vào sự biểu diễn cụ thể và cấu trúc *dữ liệu*. Hoare đã có đóng góp quan trọng để chỉnh lại các thuật ngữ và khái niệm về cấu trúc dữ liệu, qua bài "Một số điểm về cấu trúc dữ liệu". Rõ ràng là cần biết về thuật giải được áp dụng lên dữ liệu mới có thể có quyết định về cấu trúc dữ liệu, và ngược lại, cấu trúc và sự chọn lựa thuật giải cũng phụ thuộc rất nhiều vào cấu trúc dữ liệu được sử dụng. Tóm lại: *Xây dựng chương trình và cấu trúc dữ liệu không thể tách rời nhau.*

Có hai lí do để giáo trình này bắt đầu bằng một chương về cấu trúc dữ liệu. Thứ nhất là vì người ta thường có cảm nhận rằng dữ liệu

có trước thuật giải: ta phải có đối tượng trước khi có thể thao tác trên chúng. Thứ hai, và là nguyên nhân trực tiếp, giáo trình dựa vào cơ sở là bạn đọc đã quen với các khái niệm cơ bản về lập trình. Tuy nhiên, do thói quen, các chuyên đề về lập trình thường tập trung vào các thuật giải thao tác trên các cấu trúc dữ liệu đơn giản. Vì thế, có lẽ thích đáng khi chương mở đầu nói về cấu trúc dữ liệu.

Trong giáo trình và cụ thể là chương 1, chúng ta dùng thuật giải ngữ và lí thuyết của Hoare và thể hiện bằng ngôn ngữ lập trình (viết tắt là NNLT) PASCAL. Đó là: dữ liệu trước hết thể hiện sự trừu tượng của các hiện tượng thực tế và được biểu diễn như những cấu trúc trừu tượng không nhất thiết phải có trong các NNLT. Khi xây dựng chương trình, biểu diễn dữ liệu được chi tiết hóa dần - *cùng với sự tinh chế thuật giải* - để phù hợp với hệ thống lập trình sẵn có. Do đó chúng ta quy định một số nguyên lí cơ bản để xây dựng các cấu trúc dữ liệu, gọi là các *cấu trúc cơ bản*. Điều quan trọng là việc xây dựng này dễ dàng cài đặt trên máy tính thực sự. Đó là cấu trúc *mảng* (*array*), *mẩu tin* (*record*) và *tập hợp* (*set*). Không có gì ngạc nhiên, khi ba nguyên lí xây dựng cơ bản này tương ứng với những khái niệm toán học.

Cơ sở của lí thuyết về cấu trúc dữ liệu là sự phân biệt giữa cấu trúc cơ bản và cấu trúc "cấp cao". Cấu trúc cơ bản là những thành phần - tạo thành hạt nhân - để xây dựng cấu trúc cấp cao. Các biến có cấu trúc cơ bản chỉ thay đổi giá trị, nhưng không thay đổi cấu trúc cũng như tập giá trị mà chúng có thể nhận. Kết quả là kích thước vùng nhớ chúng chiếm là không đổi. Tuy nhiên, cấu trúc cấp cao được đặc trưng bởi sự thay đổi giá trị và cấu trúc của nó trong khi xử lí chương trình. Vì thế cần có những kĩ thuật phức tạp hơn để cài đặt chúng.

Tập tin tuần tự, gọi tắt là tập tin, là trung gian trong sự phân loại này. Nó có thể thay đổi độ dài. Vì tập tin tuần tự đóng vai trò cơ bản trong hầu hết các máy tính, nên nó được xem như cấu trúc cơ bản trong chương 1.

Chương 2 đề cập đến các *thuật giải sắp xếp*. Nhiều phương pháp khác nhau được đưa ra, chúng phục vụ cùng một mục đích. Việc phân tích toán học của vài thuật giải cho thấy ưu và nhược điểm của chúng, và nó làm người lập trình nhận thức được tầm quan trọng của việc

phân tích khi chọn thuật giải cho một bài toán. Việc phân thành các phương pháp sắp xếp mảng và các phương pháp sắp xếp tập tin (thường được gọi là sắp xếp trong và sắp xếp ngoài) cho thấy ảnh hưởng to lớn của biểu diễn dữ liệu đối với việc chọn thuật giải cũng như độ phức tạp của chúng. Vì là một công cụ lí tưởng để minh họa nhiều nguyên lý của lập trình và các tình huống gặp phải trong phần lớn các ứng dụng, nên vấn đề sắp xếp được xét thật kĩ. Người ta có thể xây dựng toàn bộ chuyên đề lập trình bằng cách chọn các ví dụ về sắp xếp!

Một chủ đề khác thường bị bỏ xót trong các chuyên đề nhập môn lập chương trình là đệ quy, mặc dù chúng có vai trò quan trọng trong quan điểm của nhiều thuật giải. Do đó, chương thứ ba dành cho các *thuật giải đệ quy*. Đệ quy được minh họa là sự tổng quát hóa của lặp, và như thế nó là khái niệm quan trọng và hữu hiệu trong lập trình. Nhưng đáng tiếc là có nhiều bài giảng lập trình lại dùng các ví dụ đệ quy mà đúng ra giải quyết bằng lặp thì tốt hơn. Vì thế, chương 3 tập trung vào các ví dụ mà dùng đệ quy sẽ cho cách giải tự nhiên nhất, trong khi dùng lặp sẽ dẫn đến những chương trình *phức tạp* và *cồng kềnh*. Các loại giải thuật *lần ngược* có lẽ là một áp dụng lí tưởng của đệ quy, nhưng rõ ràng nhất để dùng đệ quy là các thuật giải thao tác trên các cấu trúc dữ liệu được định nghĩa đệ quy. Các trường hợp này được xét trong chương 4.

Chương 4 bàn đến các *cấu trúc dữ liệu động*, là các dữ liệu mà cấu trúc của nó thay đổi khi xử lí chương trình. Nó cho thấy cấu trúc dữ liệu đệ quy là một loại quan trọng trong các cấu trúc động thường dùng. Mặc dù định nghĩa bằng đệ quy là có tự nhiên và có thể trong trường hợp này, nó ít được dùng trong thực hành,. Thay vào đó, người ta dùng các biến *con trỏ*. Vì thế chương 4 xét việc lập trình với các con trỏ, xâu, và các kiểu dữ liệu phức tạp hơn. Một phần quan trọng của chương này xét đến các tổ chức cây, cụ thể là *cây tìm kiếm*. Cuối chương là một ví dụ về bảng phân tán, còn gọi là mã "băm" (hash), thường được lưu ý hơn cây tìm kiếm. Điều này cho chúng ta sự so sánh giữa hai kĩ thuật khác nhau đối với một vấn đề thường gặp.

Lập trình là một nghệ thuật xây dựng. Làm thế nào để day xây

dựng và sáng tạo? Một cách là chọn ra các nguyên lí cơ sở từ nhiều trường hợp và minh họa chúng một cách hệ thống. Nhưng lập trình lại là một lĩnh vực rộng lớn thường đòi hỏi vận dụng sáng tạo. Nên không thể tóm lại bằng việc "dạy công thức". Vì vậy phương pháp là chọn và thể hiện cẩn thận các ví dụ mẫu. Dương nhiên, mỗi người có mức độ tiếp thu khác nhau từ các ví dụ. Vì vậy cách này tùy thuộc vào sự cảm nhận, sự cần cù của sinh viên. Điều này đặc biệt đúng đối với các ví dụ dài và phức tạp. Nên không phải là ngẫu nhiên khi có các ví dụ như thế trong giáo trình. Các chương trình dài là "bình thường" trong thực tế. Chúng cũng là những bài tập về việc đọc chương trình, là điều hay bị quên so với viết chương trình. Bạn đọc được dẫn dắt từng bước khi xây dựng chương trình, được thấy các "lắt léo" khi xây dựng chương trình. Chương trình được *tinh chế từng bước*. Mặc dù thể hiện các nguyên lí của thuật giải và sự phân tích toán học của nó có thể hấp dẫn những ai thích lí thuyết, nhưng lại không lôi cuốn người thực hành. Vì thế, tác giả tuân theo nguyên tắc là luôn đưa ra chương trình cuối cùng ở dạng có thể thực hiện trên máy.

Dĩ nhiên, xuất hiện một vấn đề là phải dùng dạng nào để có thể thực hiện trên máy và cũng đủ độc lập với máy. Theo yêu cầu này thì cả những kí hiệu trừu tượng lẫn các NNLT thông dụng đều không thích hợp. Tuy nhiên NNLT PASCAL lại có sự trung hòa thích đáng, nó được xây dựng cũng với mục đích trên, nên được dùng trong giáo trình này. Những ai quen với ALGOL 60 hay PL/1 có thể dễ dàng hiểu được các chương trình PASCAL. Điều đó không có nghĩa là việc chuẩn bị trước là không có lợi. Cuốn "*lập trình hệ thống*" (của tác giả) cho ta ý tưởng nền tảng vì nó cũng dùng kí hiệu PASCAL. Tuy nhiên, đây không phải là giáo trình dạy NNLT PASCAL, bạn đọc muốn tìm hiểu xin xem [1 - 3].

MỤC LỤC