

GIỚI THIỆU

Bùi Tiến Lên

01/09/2018



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Một số quy định chung

- Sinh viên *phải làm đầy đủ* các bài tập lý thuyết và thực hành
- Sinh viên không được vắng quá 3 buổi lý thuyết và thực hành
- *Cách tính điểm chung*

Tổng điểm = 50%Lý thuyết + 30%Thực hành + 20%Đồ án

- Các thuật toán
 - Phân tích thuật toán
 - Thuật toán sắp xếp
 - Thuật toán tìm kiếm
- Các cấu trúc dữ liệu
 - Mảng & danh sách liên kết (review)
 - Ngăn xếp & hàng đợi (review)
 - Cây
- Các thuật toán nâng cao
 - Nén dữ liệu
- Cấu trúc dữ liệu nâng cao
 - Đồ thị

Cấu trúc dữ liệu và giải thuật

"Giải thuật + Cấu trúc dữ liệu = Chương trình"

Niklaus Wirth

Định nghĩa 1

Cấu trúc dữ liệu & giải thuật (*data structures & algorithms*) nghiên cứu

- Tổ chức, lưu trữ *dữ liệu*
- Xây dựng và cài đặt các *thuật toán* liên quan

Cấu trúc dữ liệu và giải thuật (cont.)

- Việc lựa chọn cấu trúc dữ liệu và giải thuật có thể tạo ra sự khác biệt cho một chương trình
 - Chạy vài giây
 - Chạy vài ngày

Cấu trúc dữ liệu và giải thuật (cont.)

Ghi nhớ

- Mỗi cấu trúc dữ liệu đều có điểm mạnh và điểm yếu
- Không có một cấu trúc dữ liệu nào tốt cho mọi bài toán
- Mỗi bài toán đều có những ràng buộc về
 - không gian lưu trữ
 - thời gian thực hiện
 - khả năng lập trình
- Chỉ sau khi phân tích bài toán cẩn thận chúng ta mới có thể biết được cấu trúc dữ liệu tốt nhất để giải quyết

Cấu trúc dữ liệu

Định nghĩa 2

- **Cấu trúc dữ liệu** (*data structure*) là cách thức tổ chức (*organizing*) và lưu trữ (*storing*) để mang lại hiệu quả khi thi hành thuật toán
- Cấu trúc dữ liệu là cách thức cài đặt các kiểu dữ liệu
- **Cấu trúc dữ liệu trong** (*internal memory data structure*)
- **Cấu trúc dữ liệu ngoài** (*external memory data structure*)

Cấu trúc dữ liệu (cont.)

- Mỗi cấu trúc dữ liệu sẽ phù hợp cho những ứng dụng cụ thể
 - Ứng dụng cơ sở dữ liệu thường sử dụng *B-tree*
 - Ứng dụng trình biên dịch thường dùng *bảng băm*
 - Ứng dụng từ điển cũng thường dùng *bảng băm*
 - Ứng dụng phân phối hàng hóa thường sử dụng *hàng đợi*

Cấu trúc dữ liệu (cont.)

- Một cấu trúc dữ liệu được xem là hiệu quả cho một ứng dụng nếu thỏa:
 - Lưu trữ đầy đủ và đúng đắn dữ liệu của ứng dụng
 - Dễ dàng truy xuất và xử lý
 - Tiết kiệm bộ nhớ

Kiểu dữ liệu là gì?

Định nghĩa 3

Kiểu dữ liệu (*data type*) $T = (V, O)$

- V là tập hợp các giá trị cho kiểu dữ liệu T
- O là tập hợp các thao tác được định nghĩa trên V

Ví dụ 1

Xét T là short int

- $V = \{-32768, \dots, 32767\}$
- $O = \{+, -, *, /\}$

Kiểu dữ liệu là gì? (cont.)

- Kiểu dữ liệu trong một ngôn ngữ lập trình được phân loại thành
 - Kiểu dữ liệu cơ bản
 - Kiểu dữ liệu có cấu trúc
 - Kiểu dữ liệu trừu tượng

Kiểu dữ liệu cơ bản

Định nghĩa 4

Các ngôn ngữ lập trình (C, C++, Java, ...) đều có sẵn các kiểu dữ liệu cơ bản cho người lập trình sử dụng

Bảng 1: Các kiểu dữ liệu cơ bản

Kiểu dữ liệu	Kích thước (byte)
bool	1
char, unsigned char	1
short, unsigned short	2
int, unsigned int	4
long, unsigned long	4
long long, unsigned long long	8
float	4
double	8

Kiểu dữ liệu có cấu trúc

Định nghĩa 5

Các ngôn ngữ lập trình đều cung cấp các công cụ để tạo ra các kiểu dữ liệu mới bằng cách kết hợp các kiểu dữ liệu cơ bản theo những cấu trúc sau

- Kiểu mảng
- Kiểu chuỗi
- Kiểu cấu trúc

Kiểu dữ liệu mảng

Định nghĩa 6

Mảng dùng để biểu diễn dữ liệu ở dạng dãy các phần tử có cùng kiểu với nhau

Kiểu dữ liệu chuỗi

Định nghĩa 7

Chuỗi ký tự là mảng một chiều mà mỗi phần tử là một ký tự và ký tự cuối cùng là ký tự NULL

Kiểu dữ liệu cấu trúc

Định nghĩa 8

- Kiểu dữ liệu cấu trúc là một nhóm các thành phần có kiểu không giống nhau và mỗi thành phần được xác định bằng tên riêng
- Kiểu dữ liệu của thành phần là kiểu cơ bản hoặc lại kiểu có cấu trúc

Kiểu dữ liệu trừu tượng

Định nghĩa 9

- **Kiểu dữ liệu trừu tượng** (*abstract data type* - ADT) là một tập hợp các giá trị, cùng các thao tác trên nó
- Mỗi thao tác trên ADT được xác định thông qua dữ liệu vào và dữ liệu ra
- Không đề cập cách thức cài đặt

Kiểu dữ liệu trừu tượng (cont.)

Ví dụ 2

- ADT stack là một tập hợp các phần tử có các thao tác chính:
 - push
 - pop
 - top
- ADT stack có thể được cài đặt bằng nhiều cách khác nhau:
 - mảng một chiều
 - danh sách liên kết đơn
 - danh sách liên kết đôi

Tài liệu tham khảo



Adelson-Velskii, G. M. and Landis, E. M. (1962).

An information organization algorithm.

In *Doklady Akademia Nauk SSSR*, volume 146, pages 263–266.



Andersson, A. and Nilsson, S. (1995).

Efficient implementation of suffix trees.

Software: Practice and Experience, 25(2):129–141.



Apostol, T. M. (1976).

Introduction to analytic number theory.

Springer.

Tài liệu tham khảo (cont.)



Bauer, F. L. and Samelson, K. (2001).

Verfahren zur automatischen verarbeitung von kodierten daten und rechenmaschine zur ausübung des verfahrens.

In Pioneers and Their Contributions to Software Engineering, pages 29–40. Springer.



Bayer, R. and McCreight, E. (2002).

Organization and maintenance of large ordered indexes.

In Software pioneers, pages 245–262. Springer.



Boyer, R. S. and Moore, J. S. (1977).

A fast string searching algorithm.

Communications of the ACM, 20(10):762–772.

Tài liệu tham khảo (cont.)



Cook, S. A. (1971).

The complexity of theorem-proving procedures.

In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM.



Cormen, T. H. (2009).

Introduction to algorithms.

MIT press.



Gonnet, G. H. and Baeza-Yates, R. (1991).

Handbook of algorithms and data structures: in Pascal and C.
Addison-Wesley Longman Publishing Co., Inc.

Tài liệu tham khảo (cont.)



Guibas, L. J. and Sedgewick, R. (1978).

A dichromatic framework for balanced trees.

In Foundations of Computer Science, 1978., 19th Annual Symposium on, pages 8–21. IEEE.



Huffman, D. A. (1952).

A method for the construction of minimum-redundancy codes.

Proceedings of the IRE, 40(9):1098–1101.



Karp, R. M. and Rabin, M. O. (1987).

Efficient randomized pattern-matching algorithms.

IBM Journal of Research and Development, 31(2):249–260.



Knuth, D. E. (1976).

Big omicron and big omega and big theta.

ACM Sigact News, 8(2):18–24.

Tài liệu tham khảo (cont.)



Knuth, D. E., Morris, Jr, J. H., and Pratt, V. R. (1977).

Fast pattern matching in strings.

SIAM journal on computing, 6(2):323–350.



Landau, E., Goodman, J. E., Bateman, P. T., and Kohlbecker, E. E. (1958).

Elementary number theory.

Chelsea Publishing Company New York.



Lempel, A. (1978).

Compression of individual sequences via variable-rate coding.

IEEE Transactions on Information Theory, 24(5):530–536.



Sedgewick, R. (2002).

Algorithms in Java, Parts 1-4, volume 1.

Addison-Wesley Professional.

Tài liệu tham khảo (cont.)



Shell, D. L. (1959).

A high-speed sorting procedure.

Communications of the ACM, 2(7):30–32.



Stout, Q. F. and Warren, B. L. (1986).

Tree rebalancing in optimal time and space.

Communications of the ACM, 29(9):902–908.



Vitter, J. S. (1987).

Design and analysis of dynamic huffman codes.

Journal of the ACM (JACM), 34(4):825–845.



Wirth, N. (1986).

Algorithms and data structures.

Prentice-Hall London et al.