Cấu trúc dữ liệu và thuật toán với C++

(Data Structures and Algorithms In C++)

Giảng viên: Phạm Đình Phong

Email: phongpd@utc.edu.vn

ĐT: 0972481813

Tổ chức môn học

- Số tín chỉ: 3 + Bài tập lớn
- Hình thức thi cuối kỳ: BTL + vấn đáp
- Đánh giá kết quả học tập cuối kỳ

Các bài kiểm tra trong quá trình học, chuyên cần	Bài tập lớn và thi vấn đáp	Tổng
50%	50%	100%

- Kiểm tra viết: 10%
- Thực hành: 30%
- Chuyên cần (điểm danh): 10%

Bài 1. Bài mở đầu

I. Mục đích môn học

1. Môn học cung cấp những kiến thức cơ bản, nền tảng về một số cấu trúc dữ liệu và một số thuật toán. Nó là cơ sở để xây dựng các hệ thống phần mềm lớn và phức tạp.

   Năm 1976, Giáo sư Niklaus Wirth (Trường Đại học kỹ thuật Zurich, Thụy Sĩ, tác giả của Ngôn ngữ lập trình PASCAL năm 1970) đã xuất bản cuốn sách nổi tiếng:

ALGORITHMS +

DATA STRUCTURES =

PROGRAMS

THUẬT TOÁN + CẤU TRÚC DỮ LIỆU = CHƯƠNG TRÌNH

I. Mục đích môn học

2. Môn học giúp sinh viên hiểu cách thức tổ chức lưu trữ dữ liệu trong bộ nhớ của máy tính và làm thế nào để sử dụng nó một cách có hiệu quả trong các chương trình.

Sử dụng những kiến thức này để xây dựng các cấu trúc dữ liệu phù hợp cho các hệ thống phức tạp khác. Tức là, sử dụng các cấu trúc dữ liệu đã có để xây dựng các cấu trúc dữ liệu phức tạp hơn

Ví dụ:

- Sử dụng cấu trúc mảng để xây dựng các cấu trúc véc-tơ, ngăn xếp, đống (heap), …
- Sử dụng cấu trúc heap để xây dựng hàng đợi ưu tiên

I. Mục đích môn học

3. Cung cấp cho sinh viên một số thuật toán cơ bản trên các cấu trúc dữ liệu
4. Sinh viên hiểu và biết phân tích thời gian, không gian (bộ nhớ) cần cho một thuật toán.

II. Thời gian biểu

1	Bài 1: Bài mở đầu (introduction) Bài 2: Ngôn ngữ lập trình C++	- Một số bài tập rèn luyện kỹ năng lập trình với ngôn ngữ C++
2	Bài 3: Lập trình hướng thủ tục và lập trình hướng đối tượng (Function Oriented Programming and Object Oriented Programming) - Xây dựng lớp trong C++	- Xây dựng một số lớp đơn giản: lớp Time, Date, student,…
3	Bài 4. Xây dựng lớp mẫu, thiết kế mẫu (design pattern) trong C++.	- Làm một số lớp mẫu theo yêu cầu - Sử dụng các lớp đó trong chương trình cụ thể
4	Bài 5. Phân tích các thuật toán (Analysis of Algorithms)	-Phân tích một số thuật toán được đưa ra -Chứng minh một số thuật toán

5	Bài 6. Thuật toán đệ qui (recursive algorithm)	- Xây dựng thuật toán đệ qui giải một số bài toán
6	Bài 7. Vector	-Xây dựng lớp mẫu Vector -Xây dựng chương trình sử dụng Vector để lưu trữ dữ liệu
7	Bài 8. Danh sách liên kết đơn (single list), danh sách liên kết kép (double list)	-Xây dựng lớp mẫu danh sách liên kết đơn -Xây dựng chương trình sử dụng danh sách liên kết đơn để lưu trữ dữ liệu -Xây dựng lớp mẫu danh sách liên kết kép -Xây dựng chương trình sử dụng danh sách liên kết kép để lưu trữ dữ liệu
8	Bài 9. Cấu trúc dữ liệu kiểu ngăn xếp – Stack Bài 10. Cấu trúc dữ liệu kiểu hàng đợi – Queue	-Xây dựng lớp mẫu Stack -Xây dựng chương trình sử dụng stack để lưu trữ dữ liệu -Xây dựng lớp mẫu Queue -Xây dựng chương trình sử dụng Queue để lưu trữ dữ liệu
9	**-Kiểm tra giữa kỳ** **-Giao bài tập lớn**	
10	Bài 11. Cây nhị phân (binary tree) Bài 12. Cây tổng quát (genaral tree)	- Xây dựng lớp mẫu cây nhị phân

11	Bài 13. Đồ thị và các thuật toán trên đồ thị (Graph)	- Xây dựng lớp đồ thị với các phương thức cho phép chuyển đổi giữa các dạng lưu trữ khác nhau của đồ thị, tìm đường đi ngắn nhất giữa hai đỉnh trong đồ thị, kiểm tra đồ thị có liên thông không?,…
12	Bài 14. Các thuật toán sắp xếp (n^2) -Nổi bọt (Bubble Sort) -Chọn (Selection Sort) -Chèn (Insertion Sort)	- Cài đặt các hàm sắp xếp một mảng với từng phương thức
13	Bài 15. Các thuật toán sắp xếp (nlogn) -Trộn (Mergere Sort) -QuickSort (Quick Sort)	- Cài đặt các thuật toán sắp xếp cho các lớp Vector, List
14	Bài 16. Đống – Cây heap Thuật toán sắp xếp vun đống (Heap Sort)	- Cài đặt hàm sắp xếp một mảng bằng thuật toán HeapSort
15	Bài 17. Các thuật toán tìm kiếm (Search Algorithms) - Tìm kiếm tuần tự (Sequence search) - Tìm kiếm nhị phân (Binary search), cây tìm kiếm nhị phân (Binary search tree) - Tìm kiếm trên bảng băm – HashTable	- Cài đặt thuật toán tìm kiếm nhị phân cho lớp Vector, lớp List. - Cài đặt thuật toán tìm kiếm cho lớp cây nhị phân
16	**Nghiệm thu bài tập lớn**	

III. Các đối tượng nghiên cứu trong môn học

- **Các cấu trúc dữ liệu chuẩn**
  - Véc-tơ (Vector), danh sách (list), ngăn xếp (stack), hang đợi (queue), cây (tree), đồ thị (graph), …
- **Các thuật toán chuẩn**
  - Sắp xếp (Sorting)
  - Tìm kiếm (Selection)
- **Phân tích độ phức tạp về thời gian và không gian (bộ nhớ) của các thuật toán.**
- **Những kỹ năng lựa chọn thuật toán, cấu trúc dữ liệu và cài đặt thuật toán**

IV. Các kiến thức bổ trợ cho môn học

- Ngôn ngữ lập trình C/C++
- Phương pháp lập trình hướng đối tượng (Object Oriented Programming - OOP)

V. Tài liệu tham khảo

1. Data structures and Algorithms in C++ - Michael T. Goodrich, Roberto Tamassia
2. Introduction to Algorithms - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Revest, Clifford Stein,, MIT Press
3. Thuật toán và lập trình - Lê Minh Hoàng, ĐH Sư phạm Hà Nội
4. Cấu trúc dữ liệu và giải thuật – Nguyễn Văn Long, NXB GTVT
5. Cấu trúc dữ liệu và giải thuật – Đỗ Xuân Lôi, NXB Khoa học kỹ thuật
6. Cẩm nang thuật toán (vol1 + vol2), Robert Sedgewick, NXB KHKT
7. Lập trình hướng đối tượng và C++, Phạm Văn Ất, NXB GTVT

Địa chỉ website

8. http://www.Datastructures.net

…

VI. Cấu trúc dữ liệu

- Khái niệm
  - Cấu trúc dữ liệu là một cách lưu dữ liệu trong máy tính sao cho nó có thể được sử dụng một cách hiệu quả
- Lưu ý
  - Chất lượng và hiệu năng của chương trình phụ thuộc rất nhiều vào việc chọn cấu trúc dữ liệu
  - Một cấu trúc dữ liệu được chọn cẩn thận sẽ cho phép thực hiện thuật toán hiệu quả hơn

VII. Thuật toán

- Khái niệm
  - Một dãy hữu hạn các thao tác và trình tự thực hiện các thao tác đó sao cho sau khi thực hiện dãy thao tác này theo trình tự đã chỉ ra với đầu vào xác định ta thu được kết quả đầu ra mong muốn
- Các đặc trưng của thuật toán
  - **Đầu vào**
  - **Đầu ra**
  - **Tính đúng đắn**: kết quả của thuật toán là chính xác
  - **Tính xác định**: các bước thực hiện có trình tự xác định
  - **Tính dừng**: phải cho ra kết quả sau một số hữu hạn bước
  - **Tính phổ dụng**: áp dụng cho các bài toán cùng dạng
  - **Tính khách quan**: cho kết quả như nhau khi chạy trên các máy tính khác nhau

VII. Thuật toán

- Biểu diễn thuật toán
  - Cách 1: Ngôn ngữ tự nhiên
  - Cách 2: Ngôn ngữ lưu đồ (sơ đồ khối)
  - Cách 3: Mã giả
  - Cách 4: Các ngôn ngữ lập trình như Basic, Pascal, C/C++, Java, Python, … nhưng không nhất thiết phải sử dụng đúng ký pháp của ngôn ngữ đó

VIII. Chương trình

- Khái niệm: Chương trình là tập hợp dãy các lệnh điều khiển máy tính thực hiện
- Như vậy, chương trình là một cách diễn tả thuật toán bằng một ngôn ngữ lập trình chính xác để máy (máy tính) có thể hiểu được

VIII. Chương trình

- Các bước khi viết một chương trình
  - Bước 1: dùng một chương trình soạn thảo văn bản để viết chương trình
  - Bước 2: Dịch chương trình. Chương trình dịch sẽ dịch chương trình viết bằng ngôn ngữ lập trình cụ thể (Pascal, C/C++, …) ra mã máy dưới dạng chương trình chạy (đuôi *.com, *.exe). Một số ngôn ngữ lập trình hiện đại như Java, C# chỉ được dịch sang ngôn ngữ trung gian và được dịch ra mã máy khi chạy chương trình
  - Bước 3: chạy chương trình và gỡ lỗi (nếu có)

VIII. Chương trình

- Biên dịch: dịch toàn bộ chương trình được soạn thảo bảng ngôn ngữ lập trình nào đó sang mã máy
- Thông dịch: dịch và thực hiện ngay lập tức từng dòng lệnh khi chạy chương trình
- Pascal, C/C++ là ngôn ngữ được dịch theo kiểu biên dịch. Basic, Python là ngôn ngữ được dịch theo kiểu thông dịch

VIII. Chương trình

- Khi giải quyết một bài toán cần làm và tạo thói quen với trình tự sau:

Bài toán → Tìm và xây dựng thuật toán → Viết chương trình

- Tránh cách làm sau:

Bài toán → Viết chương trình

IX. Một số mục tiêu của công nghệ phần mềm

- Tin cậy và chính xác (Reliability – correctness)
- Hữu dụng (utility)
  - Đạt được những gì mong muốn
  - Đáp ứng đúng thời điểm
- Mềm dẻo (flexibility)
  - Có khả năng mang chuyển được (Portability), tức là có thể dễ dàng mang cài đặt sang một hệ thống khác.
  - Khả năng tương thích
    - Dễ bảo trì
    - Dễ hiểu
    - Có thể sử dụng lại
- Hiệu quả (efficiency)
  - Người lập trình (không mất quá nhiều công sức cho việc lập trình)
  - Máy
    - Thời gian
    - Bộ nhớ

X. Các nguyên lý của CNPM

- Trừu tượng (Abstract): Loại bỏ các thành phần mang tính chi tiết trong định nghĩa một vấn đề nào đó
- Module hóa (Modularity): Hạn chế độ phức tạp bằng cách phân chia ra thành nhiều phần (Chia để trị).
  - Phân chia bài toán thành các module nhỏ, thực hiện giải quyết từng module
  - Kết hợp các module
- Che dấu thông tin (Information Hiding):
  - Theo nguyên lý trừu tượng
  - Không cho phép truy nhập từ bên ngoài

Hết