



Một số thuật toán và kỹ thuật nâng cao

Kỹ thuật lập trình

ThS. Đặng Bình Phương (dbphuong@fit.hcmus.edu.vn)

Nội dung

- Thuật toán sắp xếp
- Thuật toán số học
- Quy hoạch động
- Kỹ thuật cài đặt các thuật toán hay qui trình tổng quát
- Thuật ngữ và bài đọc thêm tiếng Anh



Thuật toán sắp xếp



Khái niệm

- Sắp xếp là quá trình xử lý một danh sách các phần tử để đặt chúng theo một thứ tự theo yêu cầu nào đó
- Ví dụ: danh sách trước khi sắp xếp:
 $\{1, 25, 6, 5, 2, 37, 40\}$
Danh sách sau khi sắp xếp:
 $\{1, 2, 5, 6, 25, 37, 40\} \Rightarrow$ sắp xếp giúp cho việc tìm kiếm được nhanh hơn.
- Khi khảo sát các bài toán sắp xếp, ta phải làm việc nhiều với khái niệm nghịch thế.



Khái niệm nghịch thế

- Xét một mảng các số a_0, a_1, \dots, a_n
Nếu $i < j$ và $a_i > a_j$ thì ta gọi đó là một nghịch thế.
- Nhận xét:
 - Mảng chưa sắp xếp sẽ có nhiều nghịch thế
 - Mảng đã sắp xếp không chứa nghịch thế
 - Việc sắp xếp mảng là nhằm tìm cách giảm số nghịch thế bằng cách hoán vị các cặp nghịch thế (a_i, a_j)



Một số thuật toán sắp xếp

- Nhóm thuật toán đơn giản nhưng chi phí cao:
 - Selection sort (phương pháp chọn trực tiếp)
 - Insertion sort (phương pháp chèn trực tiếp)
 - Binary Insertion sort (phương pháp chèn trực tiếp, tìm kiếm nhị phân)
 - Interchange sort (phương pháp đổi chỗ trực tiếp)
 - Bubble sort (phương pháp “nổi bọt”)
 - Shaker sort (phương pháp “lắc”, cải tiến của “nổi bọt”)



Một số thuật toán sắp xếp

- Nhóm thuật toán phức tạp nhưng hiệu quả hơn:
 - Shell sort
 - Heap sort (phương pháp “vun đồng”)
 - Quick sort (phương pháp nhanh)
 - Merge sort (phương pháp trộn)
- Nhóm thuật toán khác:
 - Radix sort (phương pháp cơ số)
 - ...



Một số nhận xét

- Các thuật toán Bubble sort, Selection sort, Insertion sort
 - Cài đặt thuật toán đơn giản.
 - Chi phí của thuật toán cao: $O(n^2)$.
- Heap sort được cải tiến từ Selection sort nhưng chi phí thuật toán thấp hơn hẳn, bằng $O(n \log_2 n)$



Một số nhận xét

- Các thuật toán Quick sort, Merge sort là những thuật toán theo chiến lược “chia để trị”:
 - Cài đặt thuật toán phức tạp
 - Chi phí thuật toán thấp: $O(n \log_2 n)$
 - Rất hiệu quả khi dùng danh sách liên kết.
 - Trong thực tế, Quick sort chạy nhanh hơn hẳn Merge sort và Heap sort.



Một số nhận xét

- Người ta chứng minh $O(n \log_2 n)$ là ngưỡng chặn dưới của các thuật toán sắp xếp dựa trên việc so sánh giá trị của các phần tử.
- Radix sort là thuật toán phát triển theo hướng khác nên vượt qua khỏi ngưỡng trên. Nó đại diện cho nhóm thuật toán sắp xếp có độ phức tạp tuyến tính.



Thuật toán số học



Thuật toán số học

- Thuật chia Euclid và áp dụng
- Thuật toán về số nguyên tố
- Tính toán số lớn



Quy hoạch động



Ý tưởng và nguyên lý

- Phương pháp quy hoạch động thường được dùng để giải các bài toán tối ưu có bản chất đệ qui, tức là việc tìm phương án tối ưu cho bài toán đó có thể đưa về tìm phương án tối ưu của một số hữu hạn các bài toán con.
- Nguyên lý “chia để trị” thường đóng vai trò chủ đạo trong thiết kế thuật toán nói chung và trong phương pháp quy hoạch động nói riêng.



Ý tưởng và nguyên lý

- Đặc điểm giải thuật đệ qui
 - Theo hướng “từ trên xuống” (top-down) tức là bắt đầu từ bài toán lớn phân rã thành nhiều bài toán con và đi giải từng bài toán con đó.
 - Việc giải từng bài toán con lại đưa về phép phân rã tiếp thành nhiều bài toán nhỏ hơn và lại đi giải tiếp bài toán nhỏ hơn đó bất kể nó đã được giải hay chưa
 - Có thể không tối ưu về mặt thời gian thực hiện và không gian nhớ.



Ý tưởng và nguyên lý

- Đặc điểm giải thuật quy hoạch:
 - Theo hướng “từ dưới lên” (bottom-up) tức là bắt đầu từ việc giải tất cả các bài toán nhỏ (bài toán cơ sở) để từ đó từng bước giải quyết những bài toán lớn hơn cho tới khi giải được bài toán lớn nhất (bài toán ban đầu).



Một số khái niệm

- Bài toán giải theo phương pháp quy hoạch động được gọi là ***bài toán quy hoạch động***.
- Công thức phối hợp nghiệm của các bài toán con để có nghiệm của bài toán lớn là ***công thức truy hồi của quy hoạch động***.
- Tập các bài toán nhỏ nhất có ngay lời giải để từ đó giải quyết các bài toán lớn hơn gọi là ***cơ sở quy hoạch động***.
- Không gian lưu trữ lời giải các bài toán con để tìm cách phối hợp chúng gọi là ***bảng phương án của quy hoạch động***.



Điều kiện cần

- Bài toán lớn phải phân rã được thành nhiều bài toán con mà sự phối hợp lời giải đó cho ta lời giải của bài toán lớn.
- Vì quy hoạch động là đi giải tất cả các bài toán con nên nếu chúng không đủ không gian vật lý lưu trữ lời giả (bộ nhớ, đĩa, ...) để phối hợp chúng thì phương pháp quy hoạch động cũng không thể thực hiện được.
- Quá trình từ bài toán cơ sở tìm ra lời giải bài toán ban đầu phải qua hữu hạn bước.



Các bước cài đặt

- **Bước 1.** Phân tích bài toán và lập công thức truy hồi (giải pháp đệ qui).
- **Bước 2.** Giải tất cả các bài toán cơ sở (thông thường rất dễ) và lưu các lời giải vào bảng phương án.
- **Bước 3.** Dùng công thức truy hồi phối hợp những lời giải của những bài toán nhỏ đã lưu trong bảng phương án để tìm lời giải của những bài toán lớn hơn và lưu chúng vào bảng phương án (từ dưới lên). Lặp lại cho tới khi tìm được lời giải của bài toán ban đầu.
- **Bước 4.** Dựa vào bảng phương án, truy vết tìm ra nghiệm tối ưu.



Một số bài toán cơ bản

- Bài toán tìm dãy con đơn điệu tăng dài nhất
- Bài toán cái ba-lô:
 - Dạng 1: Cho trước các đồ vật
 - Dạng 2: Cho trước các loại đồ vật
- Bài toán biến đổi sâu
- Bài toán nhân ma trận
- Bài toán ghép cặp
- Bài toán di chuyển



Đồ án lập trình

- Cài đặt một số bài toán cơ bản theo gợi ý (tham khảo giáo trình Kỹ thuật lập trình).



Kỹ thuật cài đặt các thuật toán, qui trình tổng quát



Tìm hiểu thêm

- Khái niệm về tái sử dụng mã nguồn.
- Giới thiệu về con trỏ hàm và thuật toán tùy biến.
- Thuật toán tìm kiếm tùy biến cho các kiểu dữ liệu và điều kiện khác nhau.
- Thuật toán sắp xếp tùy biến.
- Quy trình xử lý cho phép thay thế các thuật toán khác nhau.



Thuật ngữ và bài đọc thêm tiếng Anh



Thuật ngữ tiếng Anh

- ***bottom-up***: từ dưới lên
- ***divide and conquer***: chia để trị
- ***dynamic programming***: quy hoạch động
- ***optimal***: tối ưu
- ***sort***: sắp xếp.
- ***swap***: hoán vị.
- ***top-down***: từ trên xuống



Bài đọc thêm tiếng Anh

- **Theory and Problems of Fundamentals of Computing with C++**, John R. Hubbard, Schaum's Outlines Series, McGraw-Hill, 1998.



