BÁO CÁO BÀI TẬP GIẢI THUẬT NGÀY 16/04/2023

# Quy trình đánh giá độ phức tạp thuật toán <Lý thuyết>

1. Phương pháp: Ước lượng phép so sánh hoán đổi của khối if
2. Thuật toán: sắp xếp chọn, chèn, nhanh

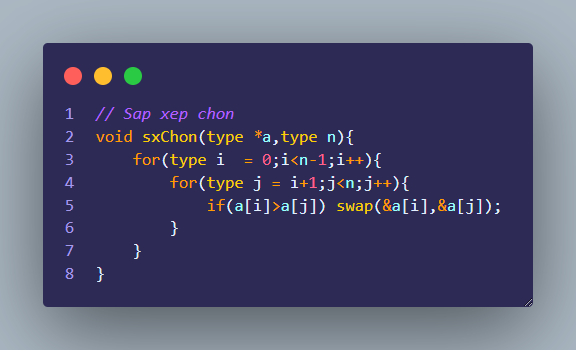
# Thực hành đánh giá

## Thuật toán sắp xếp chọn

Ý tưởng: Thuật toán đưa giá trị nhỏ nhất của dãy đang xét về đầu dãy

Phương pháp: Thuật toán sử dụng hai vòng for lồng nhau; Vòng for ngoài để thu hẹp dần dãy số cần xét, vòng for trong để đưa phần tử nhỏ nhất về đầu dãy cần xét

Code:



Đánh giá:

* Vòng for trong thuật toán sử dụng kết quả của vòng for ngoài, với mỗi kết quả thứ i trong trường hợp xấu nhất thì số câu lệnh if thực thi sẽ là n – i với i tăng dần từ 0 => n – 1. Khi đó tổng số phép thực thi của vòng for thứ 2 là khi sử dụng kết quả vòng for thứ nhất là:

=> độ phức tạp là O()

=> Tổng độ phức tạp của cả thuật toán là O()

## Thuật toán sắp xếp chèn

Ý tưởng: Thuật toán đưa giá trị mới vào đúng vị trí của nó trong dãy đang xét

Phương pháp: Thuật toán sử dụng hai vòng for lồng nhau; Vòng for ngoài để thu hẹp dần dãy số cần xét, vòng for/while trong để đưa phần tử mới chèn vào vị trí của đoạn đã sắp xếp trước

Code:

A picture containing text

Description automatically generated

Đánh giá:

* Vòng for trong thuật toán sử dụng kết quả của vòng for ngoài, với mỗi kết quả thứ i trong trường hợp xấu nhất thì số câu lệnh if thực thi sẽ là n – i với i tăng dần từ 0 => n – 1 để tìm được vị trí thích hợp, trong trường hợp dãy lấy được từ vòng for ngoài đã được sắp xếp thì số lần hoán đổi sẽ là 0. Khi đó tổng số phép so sánh của vòng for thứ 2 là:

=> độ phức tạp là O() trong trường hợp xấu nhất

=> độ phức tạp là O(n) trong trường hợp dãy đã sắp xếp

=> Tổng độ phức tạp của cả thuật toán là O() trong trường hợp xấu nhất và O(n) trong trường hợp tốt nhất

## Thuật toán sắp xếp nhanh

Ý tưởng: Thuật toán sắp xếp nhanh sử dụng phương pháp chia để trị, tức là chia dãy cần sắp xếp thành các phần nhỏ hơn để sắp xếp độc lập, sau đó ghép các phần nhỏ này lại với nhau.

Phương pháp: Trong thuật toán này, mỗi lần thực hiện phép chia dãy, ta chọn một phần tử trung vị và đưa các phần tử nhỏ hơn nó sang trái, các phần tử lớn hơn nó sang phải. Sau đó ta đệ quy sắp xếp hai phần dãy mới này.

Code:



Đánh giá:

* Độ phức tạp của thuật toán sắp xếp nhanh phụ thuộc vào sự lựa chọn của phần tử trung vị và cách chia dãy. Trong trường hợp tốt nhất, phần tử trung vị là phần tử trung tâm của dãy, và mỗi lần chia dãy đều cho ra hai phần bằng nhau. Trong trường hợp này, độ phức tạp của thuật toán là O(n log n)
* Cụ thể ta có thể xem xét quá trình chia để trị của thuật toán. Trong trường hợp tốt nhất, mỗi lần chia dãy đều chia ra hai phần bằng nhau, và thuật toán sẽ thực hiện log n lần để chia hết dãy thành các phần tử đơn lẻ. Mỗi lần chia dãy, thuật toán sẽ duyệt qua tất cả các phần tử trong dãy, nên tổng số phép so sánh sẽ là n. Do đó, tổng số phép so sánh của thuật toán sẽ là n\*log n.
* Trong trường hợp xấu nhất, nếu phần tử trung vị là phần tử nhỏ nhất hoặc lớn nhất trong dãy, hoặc dãy đã được sắp xếp theo thứ tự tăng dần hoặc giảm dần, thì thuật toán sẽ không thực hiện được phép chia đôi dãy và sẽ chỉ giảm được một phần tử. Trong trường hợp này, độ phức tạp của thuật toán là O O()

=> độ phức tạp của thuật toán sắp xếp nhanh là O(n log n) trong trường hợp trung bình và O() trong trường hợp xấu nhất.

# Quy trình đánh giá độ phức tạp thuật toán <Thực nghiệm>

1. Phương pháp: Sử dụng bộ test cá nhân được sinh bằng rand() của c, tính trung bình kết quả.
2. Bộ test: 1000, 10000, 100000 phần tử, mỗi loại số lượng phần tử sẽ có 3 bộ test khác nhau.

# Thực hành đánh giá

Kết quả:



Đánh giá:

* Với bộ test 1000: Số lượng phần tử ít, thời gian test của cả 3 thuật toán là gần như bằng nhau, thuật toán sắp xếp chọn có biểu hiện chậm hơn 2 thuật toán trên.
* Với bộ test 10000: Kết quả lúc này có sự khác biệt tương đối, bộ test chèn và chọn có thời gian sắp xếp có chênh lệch, cụ thể là sắp xếp chèn nhanh hơn gấp 3 lần so với sắp xếp chọn, còn sắp xếp nhanh vẫn chưa thể đánh giá được.
* Với bộ test 100000: Kết quả lúc này hiển thị rõ ràng hơn, sắp xếp chèn nhanh hơn sắp xếp nhanh 5 lần ở bộ test này, trong khi đó sắp xếp nhanh với pivot ở giữa thể hiện độ ưu việt với độ phức tạo O(nlogn) nhanh hơn thuật toán chèn đến hơn 300 lần.