

Báo cáo thực nghiệm Thuật toán sắp xếp

Tổng quan

Báo cáo được thực hiện nhằm đánh giá tốc độ xử lý của các thuật toán sắp xếp phổ biến, bao gồm Quick Sort, Merge Sort, Heap Sort và công cụ sắp xếp có sẵn của C++ (`std::sort`).

Cả bốn thuật toán trên đều được đánh giá có độ phức tạp là $O(n \log_2 n)$ với n là số lượng phần tử cần được sắp xếp. Tuy nhiên, do bốn thuật toán sắp xếp trên dựa trên tư tưởng khác nhau, đồng thời có nhiều điểm cần lưu ý trong khâu cài đặt, do đó chênh lệch về thời gian thực thi là điều không tránh khỏi.

Việc đánh giá thời gian thực thi của bốn thuật toán sẽ giúp đánh giá và lựa chọn được thuật toán sắp xếp tối ưu.

Dữ liệu, mã nguồn và các tài liệu có liên quan đến báo cáo này có thể được tìm thấy tại <https://github.com/vinhvien323/sort>.

Môi trường, dữ liệu và mã nguồn

Môi trường:

- Hệ điều hành: Ubuntu 20.04
- Trình biên dịch: GNU g++ 9.3.0
- CPU: Intel Core i5 - 8250U
- RAM: 12GB

Dữ liệu:

Mỗi bộ test dùng cho việc thực nghiệm là một dãy số thực có 10^6 phần tử, trong đó mỗi phần tử có giá trị tuyệt đối dưới 10^9 và có độ chính xác đến 9 chữ số sau dấu chấm thập phân.

Dữ liệu bao gồm 10 bộ test, bộ test thứ nhất được sắp xếp sẵn theo thứ tự tăng dần, bộ test thứ hai được sắp xếp sẵn theo thứ tự giảm dần, các bộ test còn lại có thứ tự sắp xếp sẵn là ngẫu nhiên.

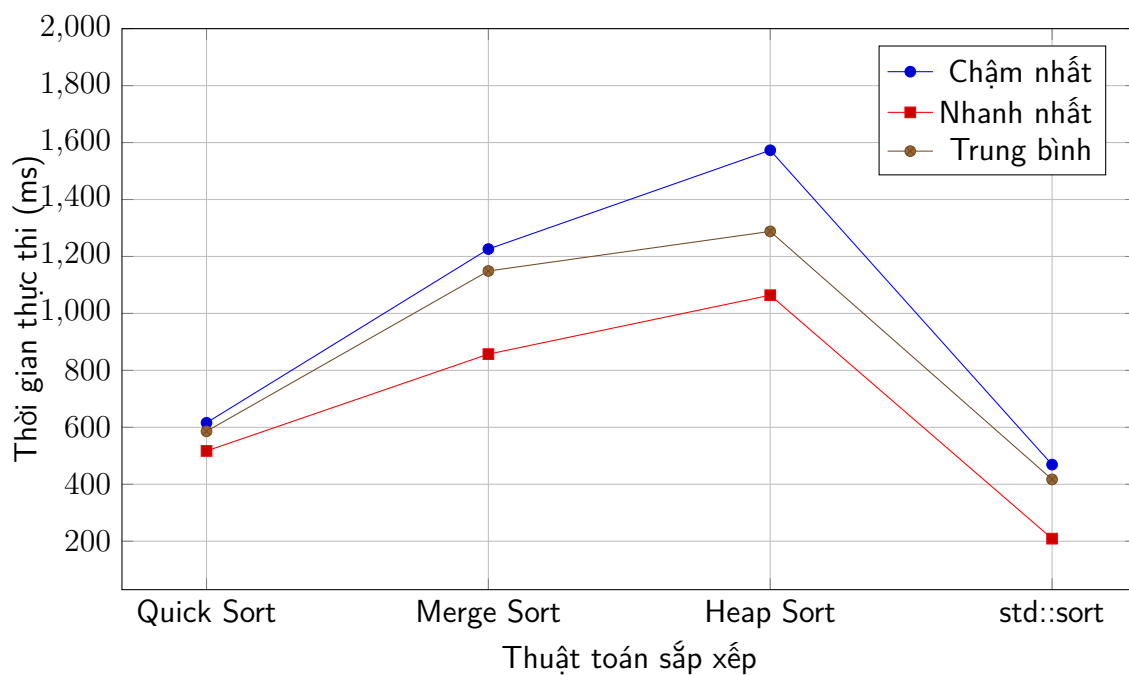
Mã nguồn:

Mã nguồn được viết bằng ngôn ngữ lập trình C++, phiên bản C++17. Kiểu dữ liệu được sử dụng để lưu trữ các phần tử là **long double**.

Kết quả thực nghiệm

	Quick Sort	Merge Sort	Heap Sort	std::sort
test01	525	874	1573	274
test02	517	857	1064	209
test03	598	1217	1277	462
test04	613	1217	1281	458
test05	616	1226	1278	458
test06	591	1217	1273	461
test07	591	1217	1276	469
test08	595	1223	1276	459
test09	602	1221	1293	465
test10	615	1217	1290	458
Chậm nhất	616	1226	1573	469
Nhanh nhất	517	857	1064	209
Trung bình	586	1149	1288	417

Bảng ghi nhận thời gian thực thi (ms)



So sánh thời gian thực thi tương ứng với các thuật toán sắp xếp

Kết luận

Trong bốn thuật toán sắp xếp được đề cập, `std::sort` có thời gian thực thi nhanh nhất và Heap Sort có thời gian thực thi chậm nhất.

Thuật toán Quick Sort và `std::sort` có thời gian thực thi gần như tương đương nhau, trừ trường hợp dãy đã cho có tính đơn điệu thì `std::sort` giảm được thời gian thực thi khá đáng kể.

Merge Sort và Heap Sort có tốc độ chạy chậm hơn đáng kể do sử dụng cấu trúc dữ liệu phức tạp và khó cài đặt.

Tổng kết: `std::sort` được khuyến nghị sử dụng trong hầu hết trường hợp do có thời gian thực thi vượt trội và đã được cung cấp sẵn trong ngôn ngữ lập trình.