

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 01/03 – 16/03/2022

Sinh viên thực hiện:

- Họ Và Tên: Trần Hoàng Bảo Ly
- MSSV: 21521109

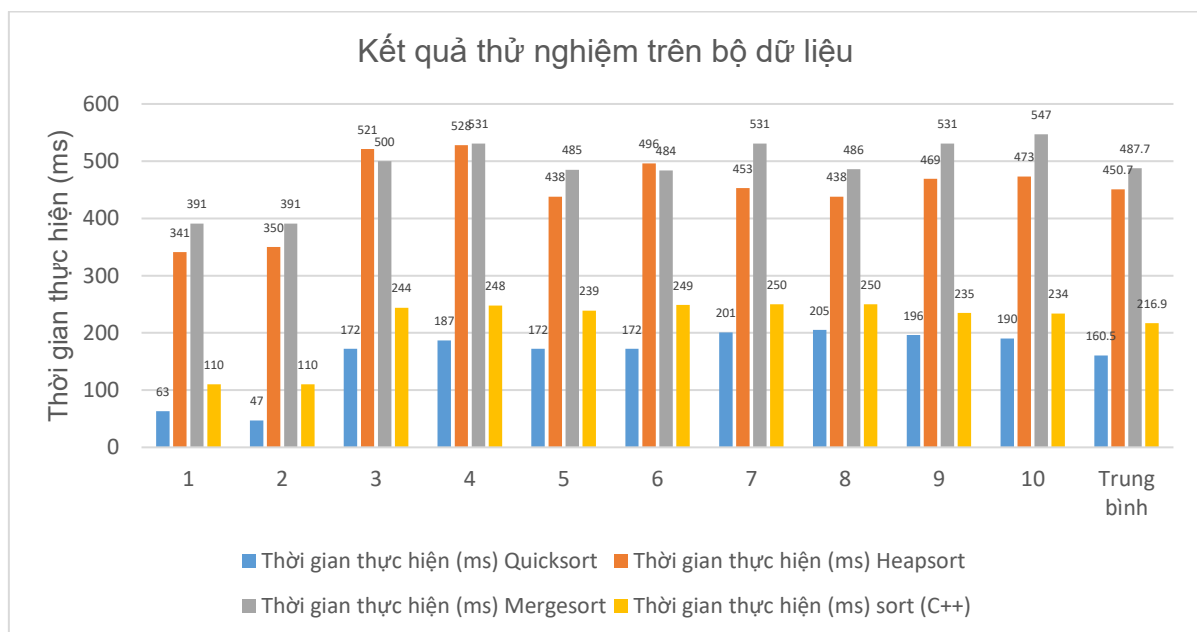
Nội dung báo cáo:

I. Kết quả thử nghiệm

1. Bảng thời gian thực hiện¹

Dữ liệu	Thời gian thực hiện (ms)			
	Quicksort	Heapsort	Mergesort	sort (C++)
1	63	341	391	110
2	47	350	391	110
3	172	521	500	244
4	187	528	531	248
5	172	438	485	239
6	172	496	484	249
7	201	453	531	250
8	205	438	486	250
9	196	469	531	235
10	190	473	547	234
Trung bình	160.5	450.7	487.7	216.9

2. Biểu đồ (cột) thời gian thực hiện



¹ Số liệu chỉ mang tính minh họa

II. Kết luận:

Cả 10 bộ test trên đều có 10^6 Phần tử là các số thực trong khoảng từ 0 đến 10^6 . Trong 10 bộ test đầu tiên có 2 bộ test đầu tiên nằm trong trường hợp đặc biệt. Bộ test đầu tiên đã được sắp xếp theo chiều tăng dần, còn bộ test thứ 2 được sắp xếp theo chiều giảm dần. Nhìn chung cả 4 thuật toán sắp xếp đều có xu hướng xử lý nhanh hơn với hai bộ test đặc biệt trên. (Ban đầu thuật toán Quick Sort thực hiện chọn pivot là điểm cuối, với hai test đầu, chương trình đã gặp phải trường hợp tệ nhất nhất với độ phức tạp $O(n^2)$, nhưng sau khi cải tiến cách chọn pivot ở điểm giữa, thuật toán Quick Sort đã tránh được các trường hợp trên).

Cả 4 thuật toán trên đều có độ phức tạp là $O(n \log n)$ tuy nhiên thuật toán Mersort có thời gian thực hiện trung bình chậm nhất với 487.7 ms, tiếp đến là HeapSort với thời gian thực hiện trung bình là 450.7 ms, ở một vài trường hợp ví dụ test 3, test 6 thuật toán MergeSort lại có tốc độ nhanh hơn.

Thuật toán QuickSort có thời gian thực hiện nhanh nhất khi tốc độ trung bình của nó là 160.5 ms gấp 3.03 lần so với thuật toán MergeSort và gấp 2.81 lần so với thuật toán HeapSort. Theo sau là `std::sort` với tốc độ trung bình là 216.9 ms chậm hơn 26% so với thuật toán QuickSort.

- ⇒ Tùy thuộc vào bộ dữ liệu khác nhau tốc độ thực hiện của các thuật toán sắp xếp cũng khác nhau. Cần phải cân nhắc và lựa chọn kỹ càng các thuật toán phù hợp khi thực hiện.
- ⇒ Khi thiết kế chương trình thiên về tốc độ sắp xếp. Ưu tiên sử dụng thuật toán QuickSort khi thực hiện sắp xếp. Ngược lại sử dụng `std::Sort` đơn giản và dễ thực hiện hơn.
- ⇒ Các thuật toán còn lại tùy trường hợp và yêu cầu của hệ thống có thể chọn để cài đặt một cách đơn giản.

III. Thông tin chi tiết – link github, trong repo gibub cần có

1. Báo cáo
2. Mã nguồn
3. Dữ liệu thử nghiệm

Github link: <https://github.com/ONGDOGA1/SortTime-Report>.