

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Sinh viên thực hiện: Nguyễn Thanh Quỳnh Anh - 25520085

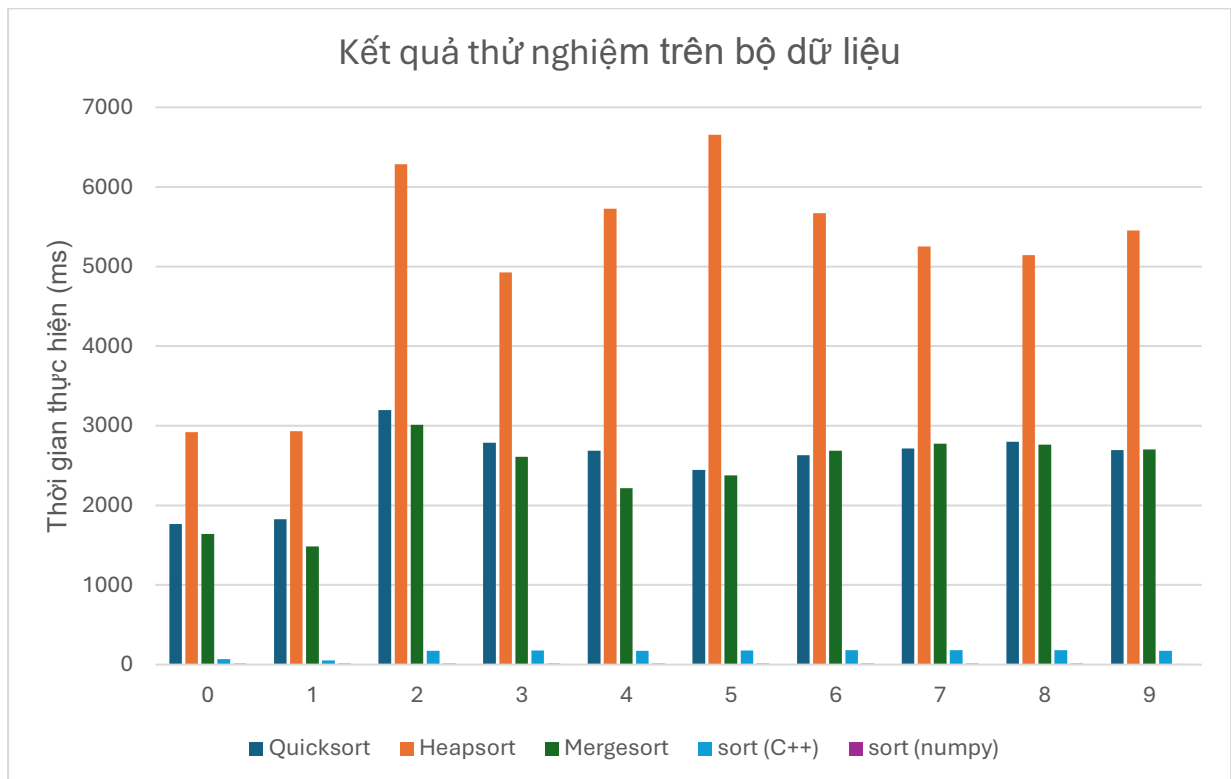
Nội dung báo cáo:

I. Kết quả thử nghiệm

1. Bảng thời gian thực hiện¹

Dữ liệu	Thời gian thực hiện (ms)				
	Quicksort	Heapsort	Mergesort	sort (C++)	sort (numpy)
0	1767.38	2921.39	1639.71	70.2032	13.23
1	1827.57	2929.79	1483.40	52.712	12.12
2	3195.13	6283.15	3011.12	175.167	11.58
3	2786.64	4926.79	2609.21	176.221	13.38
4	2687.75	5726.31	2217.18	174.169	13.77
5	2444.02	6655.80	2375.49	177.462	10.75
6	2632.00	5670.88	2684.65	183.167	12.71
7	2716.44	5250.14	2775.05	183.083	12.19
8	2800.44	5144.27	2764.15	180.189	12.31
9	2694.26	5452.57	2704.23	173.421	10.06
Trung bình	2555.16	5096.11	2426.42	154.58	12.21

2. Biểu đồ (cột) thời gian thực hiện



¹ Số liệu chỉ mang tính minh họa

II. Kết luận:

- **Numpy nhanh nhất:** Thời gian trung bình chỉ **12.21 ms**. Nó nhanh hơn các thuật toán tự viết khoảng 200 lần. Lý do là Numpy được tối ưu hóa cực mạnh bằng ngôn ngữ máy cấp thấp.
- **Hàm sort của C++ đứng nhì:** Với khoảng **154.58 ms**, hàm dựng sẵn của C++ rất nhanh và ổn định, vượt xa các thuật toán truyền thống tự cài đặt.
- **Nhóm thuật toán tự cài đặt (Quicksort, Mergesort, Heapsort):**

Mergesort (2426.42 ms): Tốc độ ổn định nhất trong nhóm tự viết.

Quicksort (2555.16 ms): Tốc độ xấp xỉ Mergesort, hoạt động tốt trên cả dãy đã sắp xếp lẫn dãy ngẫu nhiên.

Heapsort (5096.11 ms): Chậm nhất. Thời gian thực thi lâu gấp đôi so với Quicksort và Mergesort.

- **Ảnh hưởng của loại dữ liệu:**
 1. Các thuật toán đều chạy nhanh hơn trên dãy số đã có thứ tự (Dữ liệu 0 và 1) so với dãy số ngẫu nhiên.
 2. Với dãy số ngẫu nhiên (Dữ liệu 2-9), thời gian thực thi của mỗi thuật toán khá đồng đều, không có biến động bất thường.

III. Thông tin chi tiết – link github:

<https://github.com/quynhanhnguyenthanh/Sorting-Algorithms-Experiment>