Lớp: IT003.P21.CTTN

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 25/02 – 27/02/2025

Sinh viên thực hiện: Phan Thị Như Huỳnh - 24520717

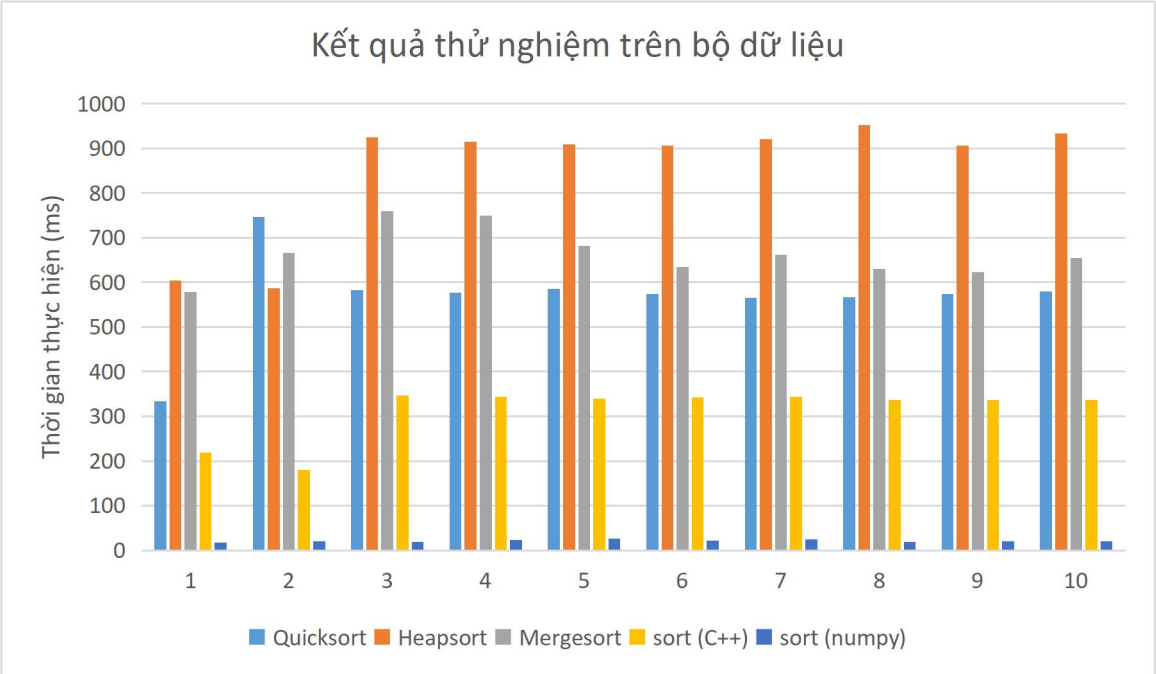
Nội dung báo cáo:

I. Kết quả thử nghiệm

1. Bảng thời gian thực hiện[1]

Dữ liệu	Thời gian thực hiện (ms)				
	Quicksort	Heapsort	Mergesort	sort (C++)	sort (numpy)
1	334.046	604.946	578.992	218.042	17.305
2	746.973	587.027	666.043	180.341	20.622
3	582.368	924.86	759.373	347.225	18.627
4	576.634	915.703	749.144	344.028	22.506
5	585.384	909.855	682.666	340.023	25.451
6	574.742	905.93	634.267	341.868	21.371
7	564.795	920.847	662.04	343.56	24.715
8	567.386	952.831	629.819	336.568	18.786
9	574.173	906.13	623.366	336.994	20.099
10	579.224	934.469	655.26	336.518	20.378
Trung bình	548.573	856.2598	664.097	312.517	20.986

2. Biểu đồ (cột) thời gian thực hiện



[1] Số liệu chỉ mang tính minh họa

II. Kết luận:

- Dựa trên biểu đồ, có thể rút ra một số nhận xét chính như sau:

    - Nhìn chung, Quicksort thường có thời gian thực thi ngắn hơn so với Heapsort và Mergesort trong hầu hết các bộ dữ liệu.
    - Heapsort thường cho thời gian lớn nhất (chạy chậm nhất) trong ba giải thuật này.
    - Mergesort nằm ở mức trung bình, nhanh hơn Heapsort nhưng chậm hơn Quicksort.
    - Cả sort (C++) và sort (numpy) đều cho kết quả tốt hơn đáng kể so với ba giải thuật tự cài đặt. Giữa hai hàm này, sort (numpy) nhìn chung nhanh hơn sort (C++).

- **Kết luận**:
    - Các hàm sort có sẵn trong C++ và Python thường là lựa chọn tốt nhất về hiệu năng.
    - Nếu phải tự cài đặt giải thuật, Quicksort (với phiên bản tối ưu và chọn pivot tốt) thường cho kết quả khả quan hơn so với Mergesort và Heapsort.

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

1. Link github chung
2. Báo cáo
3. Mã nguồn
4. Dữ liệu thử nghiệm