

Lớp: ATTN_2025-IT003.Q21.CTTN

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Sinh viên thực hiện: Vũ Thị Thu Hương MSSV: 25520671

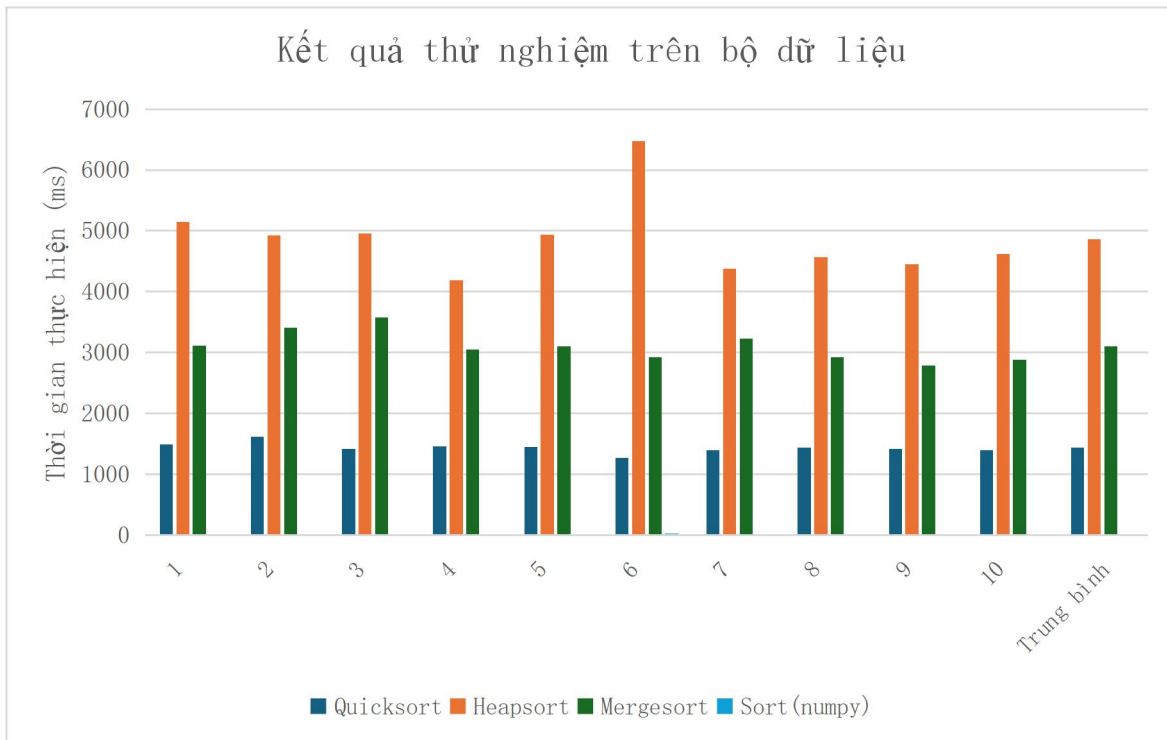
Nội dung báo cáo:

I. Kết quả thử nghiệm

1. Bảng thời gian thực hiện

Dữ liệu	Thời gian thực hiện (ms)			
	Quicksort	Heapsort	Mergesort	Sort(numpy)
1	1492.5987	5145.471	3114.1667	15.0797
2	1617.2118	4930.6073	3407.3143	14.9113
3	1412.6013	4956.9983	3574.0158	14.2326
4	1461.4436	4193.1393	3050.2614	12.1195
5	1443.4117	4940.8803	3106.4829	13.7739
6	1269.0986	6470.7023	2924.2851	9.3495
7	1397.0563	4377.033	3226.3848	12.9662
8	1436.51	4566.027	2923.3253	11.4001
9	1410.5049	4446.4762	2781.4909	10.6512
10	1395.6299	4616.8018	2879.1824	12.5143
Trung bình	1433.60668	4864.41365	3098.69096	12.6998

2. Biểu đồ (cột) thời gian hiển thị



II. Kết luận:

- sort() của NumPy nhanh nhất

==> Thời gian trung bình chỉ khoảng **12.6998 ms**, nhỏ hơn rất nhiều so với 3 thuật toán tự cài.

- So sánh theo trung bình:

+ Nhanh hơn **QuickSort ~ 113 lần**

+ Nhanh hơn **MergeSort ~ 244 lần**

+ Nhanh hơn **HeapSort ~ 383 lần**

==> Điều này cho thấy hàm sort của NumPy đã được tối ưu rất mạnh.

- Trong 3 thuật toán tự cài, **QuickSort nhanh nhất**

+ Trung bình:

QuickSort: ~1433.61 ms (nhanh nhất)

MergeSort: ~3098.69 ms (đứng thứ 2)

HeapSort: ~4864.41 ms (chậm nhất)

==> Với bộ dữ liệu thử nghiệm QuickSort cho hiệu năng tốt nhất.

- HeapSort chậm nhất và dao động mạnh

+ HeapSort luôn cao nhất, đặc biệt ở **dữ liệu số 6 (~6470.70 ms)** tăng vọt rõ rệt.

==> Điều này cho thấy HeapSort trong cài đặt hiện tại có chi phí thao tác lớn (nhiều bước heapify/đổi chỗ), nên thời gian thực thi cao và kém ổn định hơn.

- MergeSort ổn định hơn HeapSort nhưng vẫn chậm hơn QuickSort

+ MergeSort dao động quanh ~2800–3600 ms, nhìn chung “đều” hơn HeapSort.

+ Tuy nhiên vẫn chậm hơn QuickSort khá nhiều.

- Tổng kết xếp hạng tốc độ (từ nhanh → chậm)

Sort (NumPy) > QuickSort > MergeSort > HeapSort

III. Thông tin chi tiết:

<https://github.com/baht21/SORTING.git>