

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 28/02 –28/02/2025

Sinh viên thực hiện:

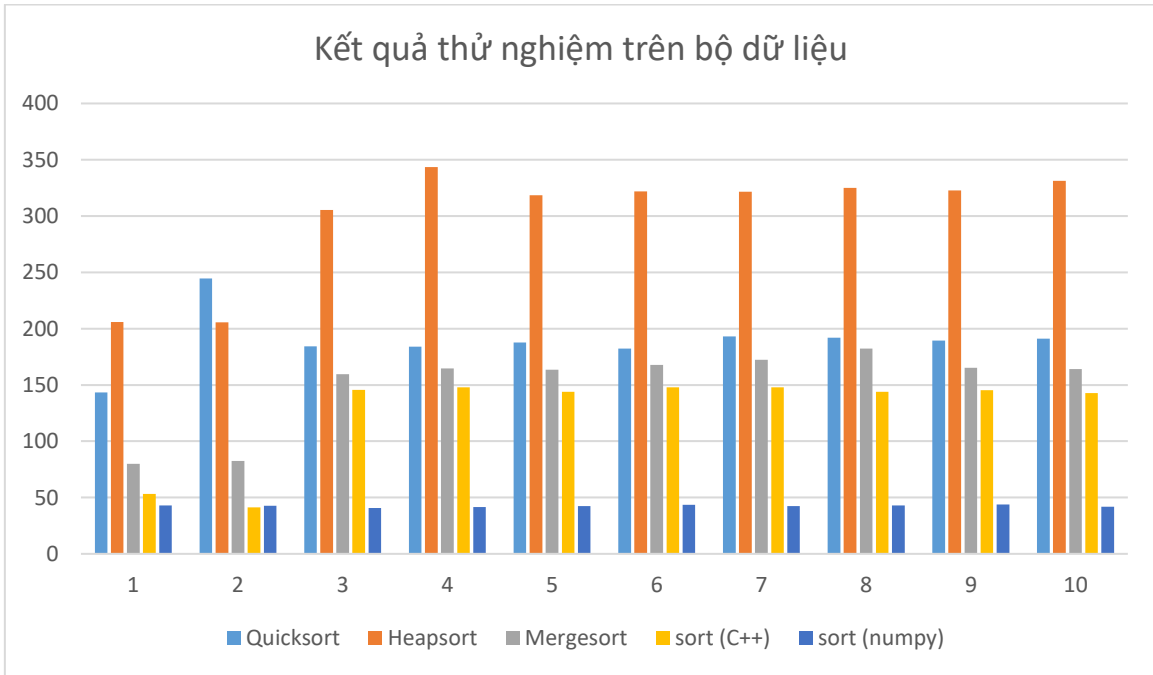
Nội dung báo cáo:

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)
1	143.284	205.942	79.836	53.1584	42.98
2	244.572	205.583	82.5813	41.1419	42.78
3	184.216	305.348	159.566	145.63	40.65
4	183.891	343.475	164.602	147.853	41.52
5	187.598	318.337	163.39	143.777	42.48
6	182.323	321.986	167.726	147.871	43.53
7	193.07	321.602	172.474	147.981	42.43
8	191.98	324.928	182.242	143.809	42.97
9	189.523	322.642	165.376	145.282	43.95
10	191.053	331.336	164.012	142.806	41.86
Trung bình	189.151	300.118	150.181	125.931	42.515

2. Biểu đồ (cột) thời gian thực hiện



¹ Số liệu chỉ mang tính minh họa

II. Kết luận:

- Dựa vào kết quả của biểu đồ, ta có thể kết luận :
 - Ở hầu hết tất cả các test, **Heapsort(cột màu cam)** tuy với độ phức tạp $O(N\log N)$ nhưng lại có hiệu suất kém nhất vì có nhiều thao tác trên bộ nhớ hơn so với các thuật toán khác.
 - **Quicksort(cột màu xanh nhạt)** với độ phức tạp $O(N\log N)$, nó có hiệu suất trung bình ở hầu hết các test nhưng với test 2 là một trường hợp xấu nhất của Quicksort khi dãy là một dãy giảm dần, việc không chọn khóa (pivot) đủ mạnh sẽ làm cho thuật toán chạy khá chậm.
 - **Mergesort (cột màu xám)** với độ phức tạp $O(N\log N)$, khá ổn định với tất cả các test.
 - **Sort(C++) (cột màu vàng)** với độ phức tạp $O(N\log N)$, do có sự tích hợp sẵn của nhiều thuật toán sắp xếp, Sort(C++) cho thấy sự ổn định và hiệu suất hơn so với cả Quicksort và MergeSort.
 - **Sort(numpy) (cột màu xanh đậm)** với độ phức tạp $O(N\log N)$, có hiệu suất nhanh nhất và ổn định nhất trong tất cả các test được so sánh.

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

- Link repo github : <https://github.com/MQAnh/SortingAlgorithms>