

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

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

Sinh viên thực hiện: Trần Xuân Đạt - 24520008

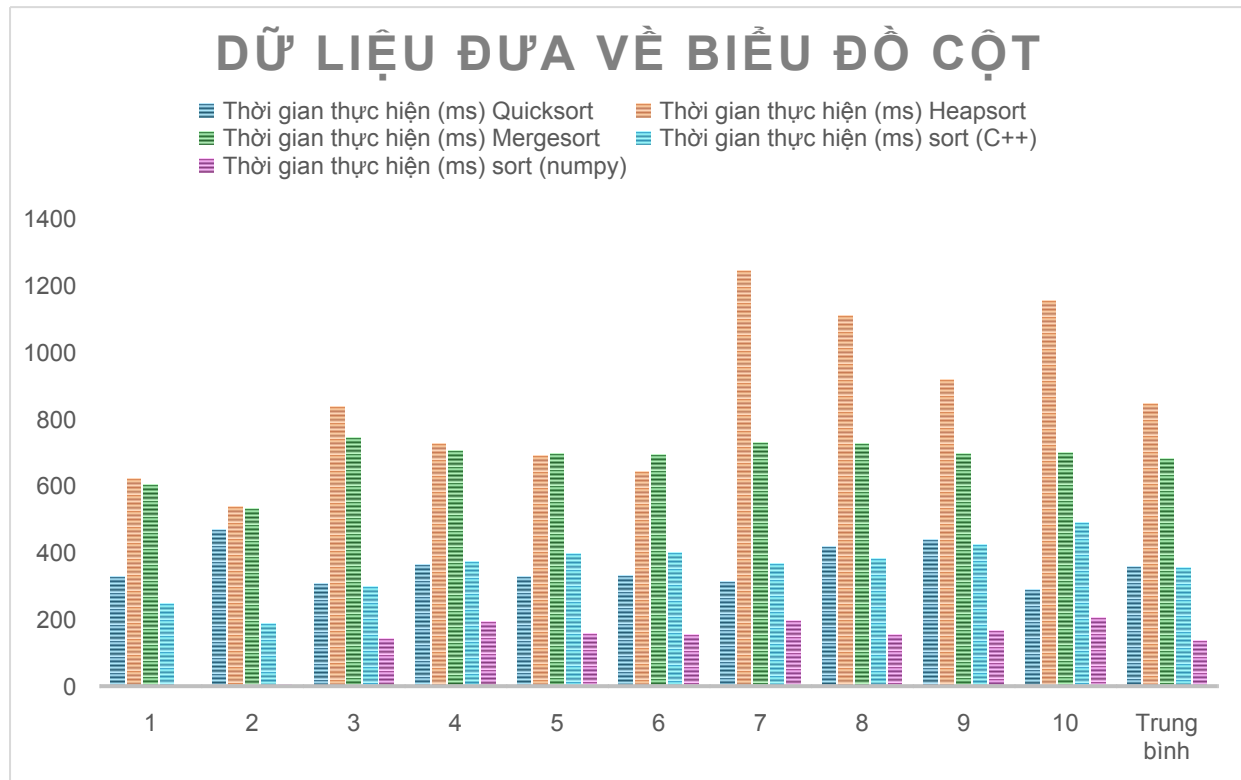
Nội dung báo cáo:

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	329	622	605	249	3
2	469	538	532	188	4
3	308	837	746	300	144
4	366	728	707	374	195
5	328	691	697	399	158
6	332	643	694	400	155
7	313	1245	731	367	197
8	419	1110	728	384	156
9	439	920	698	424	166
10	291	1155	701	492	206
Trung bình	359	848	683	357	138

2. Biểu đồ (cột) thời gian thực hiện



¹ Số liệu chỉ mang tính minh họa

II. Kết luận:

1. Hiệu suất của các thuật toán khác nhau:

Quicksort (màu xanh dương) và **Mergesort** (màu xanh lá) có thời gian thực thi tương đối tương đương, với Mergesort hơi chậm hơn. **Heapsort** (màu cam) có thời gian thực thi lâu nhất trong số các thuật toán được so sánh. **Sort (C++)** (màu xanh nhạt) là một trong những thuật toán nhanh nhất. **Sort (numpy)** (màu tím) có hiệu suất tốt nhất, thời gian thực thi thấp nhất.

2. Sự ổn định giữa các lần chạy:

Các thuật toán như Quicksort, Mergesort, và Sort (C++) có sự ổn định tương đối giữa các lần chạy khác nhau. Heapsort có sự biến động lớn hơn về thời gian thực thi.

3. Trung bình thời gian thực thi:

Heapsort có thời gian thực thi trung bình cao nhất. Sort (numpy) có thời gian thực thi trung bình thấp nhất, cho thấy đây là một lựa chọn tối ưu khi cần sắp xếp dữ liệu nhanh chóng.

Tóm lại, nếu mục tiêu là tốc độ thực thi tối ưu, **sort (numpy) hoặc sort (C++)** là những lựa chọn tốt nhất. Nếu cần một thuật toán với hiệu suất ổn định, **Quicksort hoặc Mergesort** là lựa chọn phù hợp. **Heapsort** có thể không phải là phương án tối ưu vì thời gian thực thi lâu.

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

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

Link Github: <https://github.com/DATKINGKHUNG/SortingAlgorithms>