

Lớp:

# BÁO CÁO KẾT QUẢ THỬ NGHIỆM

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

Sinh viên thực hiện:

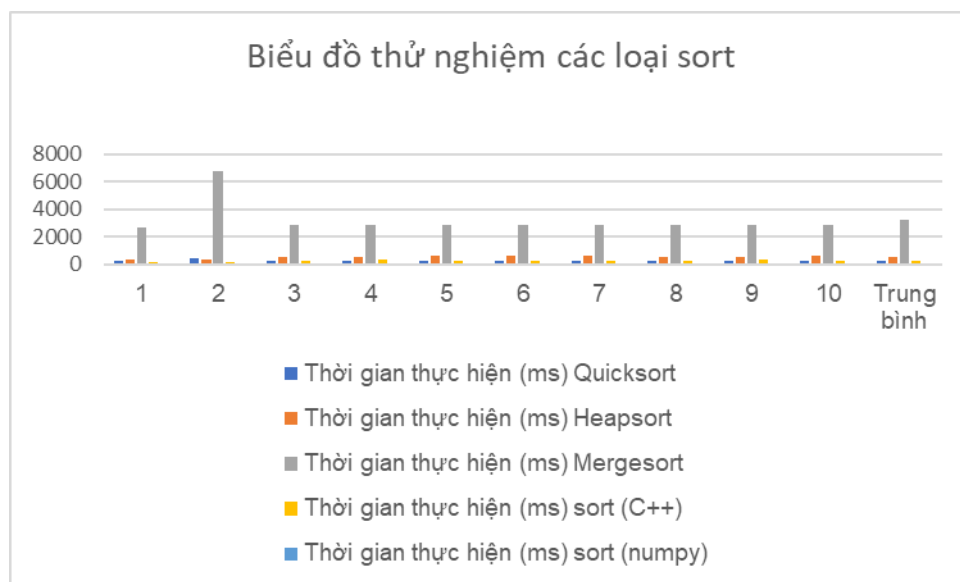
Nội dung báo cáo:

## I. Kết quả thử nghiệm

### 1. Bảng thời gian thực hiện<sup>1</sup>

Dữ liệu	Thời gian thực hiện (ms)				
	Quicksort	Heapsort	Mergesort	sort (C++)	sort (numpy)
1	214	362	2653	142	14
2	389	349	6763	125	11
3	250	561	2853	281	10
4	257	552	2811	297	10
5	252	590	2845	285	11
6	255	605	2815	280	12
7	248	573	2846	279	11
8	269	523	2829	283	11
9	261	567	2803	308	11
10	258	628	2808	286	11
Trung bình	265.3	531	3202.6	228.6	11.2

### 2. Biểu đồ (cột) thời gian thực hiện



## II. Kết luận:

- Về mặt tổng quan, quicksort và heapsort cho thời gian gần giống nhau (mặc dù mỗi cái đều có thể mạnh riêng). Merge sort thì cho ra thời gian lâu nhất (gấp 10 lần quick sort) có lẽ là do merge sort cần cấp bộ nhớ nhiều để lưu các mảng con, nếu bộ nhớ không đủ có thể dẫn tới paging, còn quicksort thì không, vậy đối với một mảng có số

<sup>1</sup> Số liệu chỉ mang tính minh họa

lượng phần tử lớn (như 1 triệu phần tử) thì đối với máy yếu, merge sort làm việc không hiệu quả. Đối với sort của python, cho ra thời gian nhanh nhất, do độ hoàn thiện và cải tiến của nó. Sort của python là Timsort, thuật toán lai giữa merge sort và insertion sort, nên nó hoạt động gần như hiệu quả trong mọi trường hợp. Đối với hàm sort có sẵn trong C++, nó sort nhanh và tối ưu hơn quicksort và heapsort, có lẽ nó là sự kết hợp của cả quicksort và heapsort. Hàm sort có sẵn trong C++ sẽ sử dụng thuật toán quicksort trước, nếu thấy đệ quy quá sâu, thì lập tức sẽ chuyển qua heapsort để tối ưu, chính vì điều đó mà trong C++, hàm sort có sẵn sort nhanh hơn hẳn quicksort và heapsort.

**Sau đây là link github :** <https://github.com/maxxduyanh/SortingAlgorithms>