

## BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 26/02/2025 – 01/03/2025

Sinh viên thực hiện: Võ Quang Vũ - 24522045

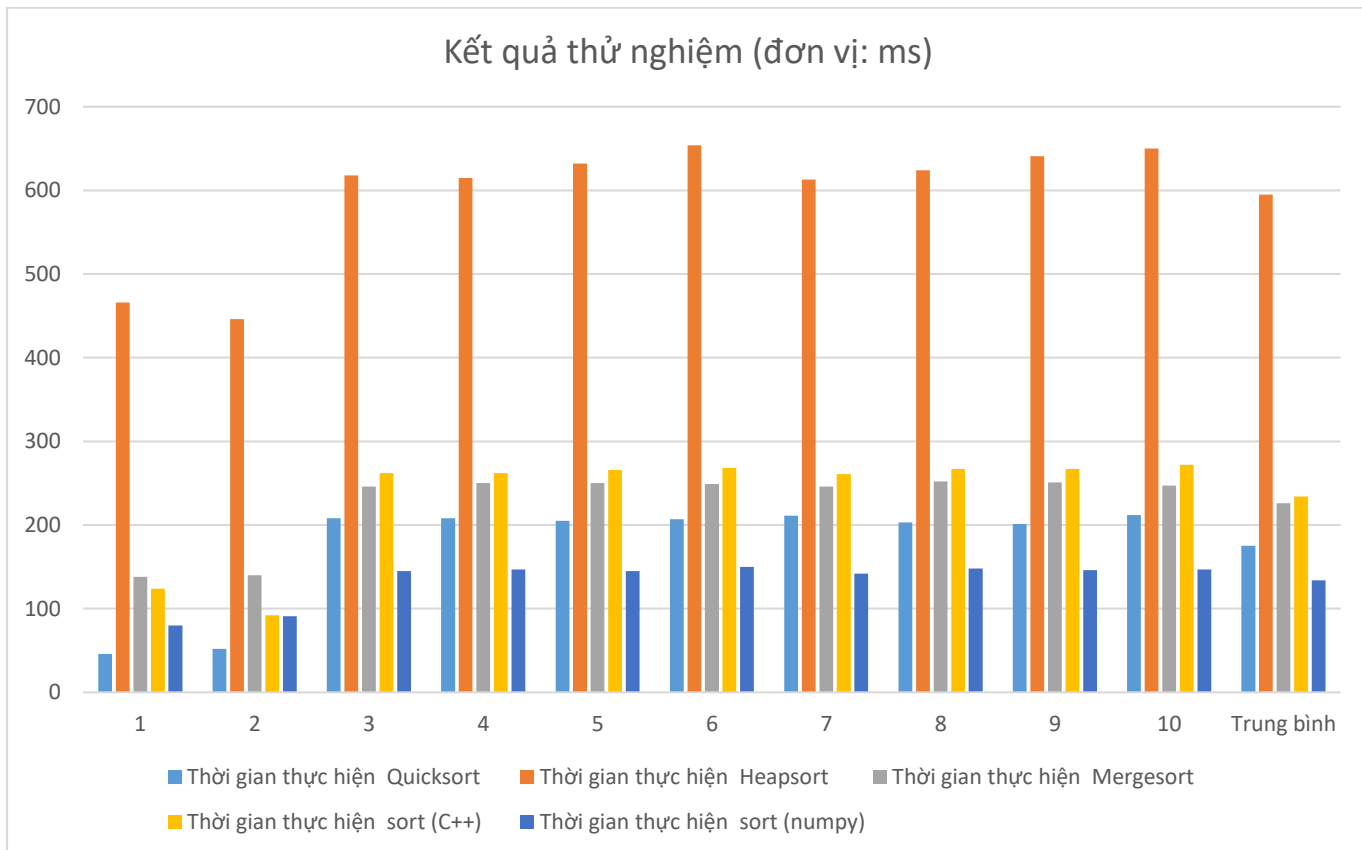
Nội dung báo cáo: Đánh giá hiệu suất các thuật toán sắp xếp trên các bộ test khác nhau và so sánh chúng

### 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	46	466	138	124	80
2	52	446	140	92	91
3	208	618	246	262	145
4	208	615	250	262	147
5	205	632	250	266	145
6	207	654	249	268	150
7	211	613	246	261	142
8	203	624	252	267	148
9	201	641	251	267	146
10	212	650	247	272	147
Trung bình	175	595	226	234	134

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



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

## II. Kết luận:

- **Heapsort có thời gian thực hiện cao nhất:** Dễ dàng nhận thấy rằng cột màu cam (Heapsort) luôn cao nhất trong tất cả các lần thử nghiệm, cho thấy thuật toán này có hiệu suất kém hơn so với các thuật toán còn lại.
  - **Quicksort và sort (C++) có hiệu suất tốt:** Cột màu xanh dương (Quicksort) và màu vàng (sort của C++) có thời gian thực hiện thấp hơn so với Heapsort và Mergesort, chứng tỏ chúng hoạt động hiệu quả hơn trong các thử nghiệm.
  - **Sort (numpy) có thời gian thực hiện thấp nhất:** Cột xanh đậm (sort của numpy) có xu hướng thấp hơn các thuật toán khác, điều này cho thấy thư viện numpy tối ưu hóa rất tốt thuật toán sắp xếp.
  - **Mergesort có hiệu suất trung bình:** Cột màu xám (Mergesort) có chiều cao trung bình, cao hơn Quicksort nhưng thấp hơn Heapsort, thể hiện rằng nó có hiệu suất ổn định nhưng không tối ưu nhất.
  - **Sự ổn định giữa các lần thử nghiệm:** Các thuật toán có xu hướng giữ mức thời gian thực hiện ổn định qua các lần thử, với sự khác biệt không quá lớn giữa các lần chạy.
- ⇒ **Tổng kết:** Nếu cần một thuật toán sắp xếp nhanh và hiệu quả, có vẻ như **sort của numpy hoặc Quicksort** là lựa chọn tốt nhất

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

*Link github:*

<https://github.com/threalwinky/IT003.P21.CTTN/tree/main/Homework2>

*Lưu ý: Chương trình của em đã được tự động hoá và có thể chạy trên Linux (file sorting) hoặc Windows (file sorting.exe)*

### 1. Kết quả

Đây là link chứa các file báo cáo về thời gian khi em chạy các thuật toán trên 5 bộ testcases khác nhau, trong các bộ testcases chứa các test theo yêu cầu bao gồm test1 tăng dần, test2 giảm dần và các test còn lại random. Kết quả của bảng thời gian thực hiện nhờ thống kê của file statistics.py để tính thời gian trung bình của 5 bộ testcases.

<https://github.com/threalwinky/IT003.P21.CTTN/tree/main/Homework2/sorting/result>

### 2. Mã nguồn

Đây là mã nguồn của các thuật toán sort mà em sử dụng để chạy bao gồm Quicksort, Heapsort, Mergesort, Sort(C++), Sort(Numpy)

<https://github.com/threalwinky/IT003.P21.CTTN/tree/main/Homework2/sorting/script>

### 3. Dữ liệu thử nghiệm

Với mỗi bộ testcases em sẽ sử dụng C++ để tạo ra 10 test thỏa yêu cầu

<https://github.com/threalwinky/IT003.P21.CTTN/blob/main/Homework2/sorting/core/generate.cpp>