

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 25/02 – 3/3

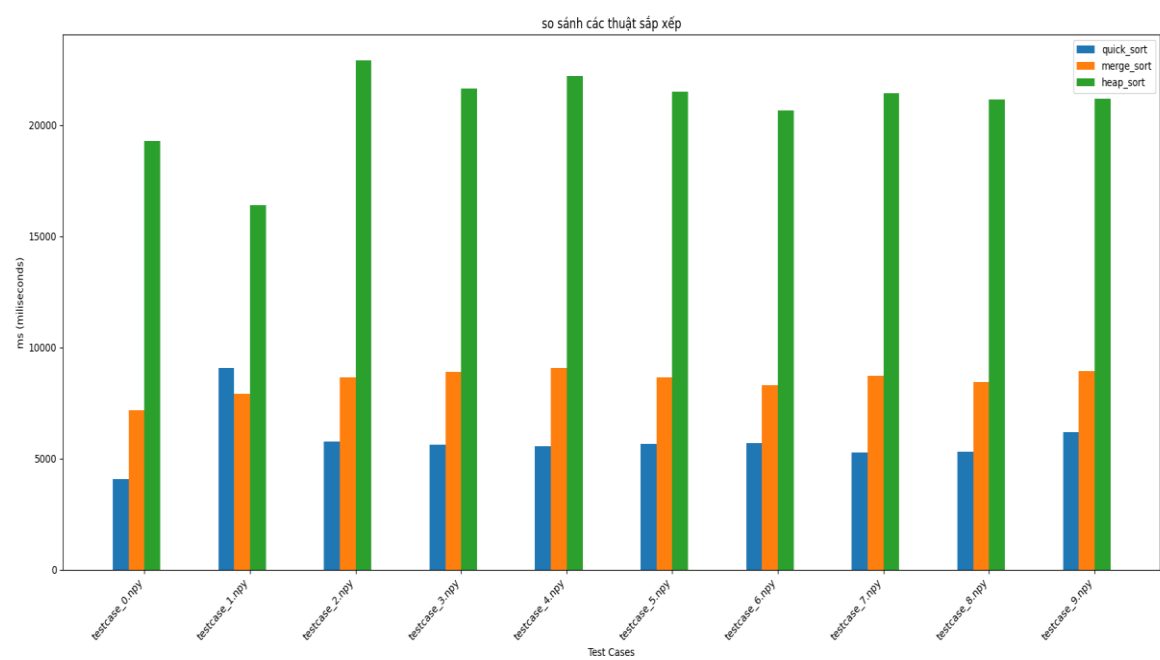
Sinh viên thực hiện: Lê Trí Đức

Nội dung báo cáo: Thực nghiệm các giải thuật sắp xếp

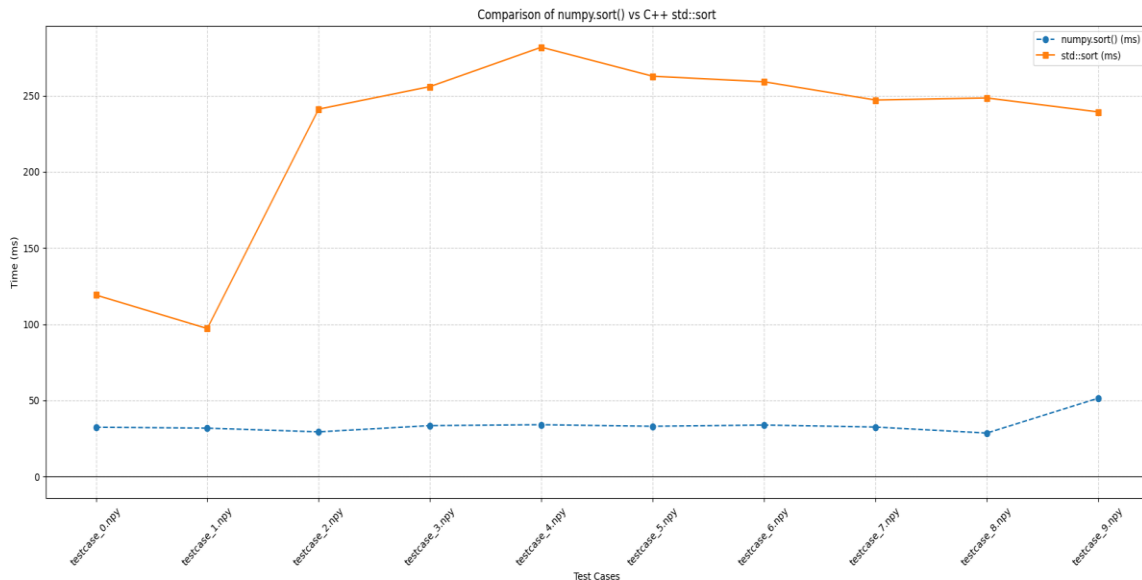
I. Kết quả thử nghiệm  
1. Bảng thời gian thực hiện<sup>1</sup>

Dữ liệu	Thời gian thực hiện (ms)				
	Quicksort	Heapsort	Mergesort	sort (C++)	sort (numpy)
1	4083	19267	7156	119	32
2	9069	16410	7921	97	31
3	5756	22923	8656	241	29
4	5607	21627	8906	255	33
5	5563	22203	9069	281	34
6	5637	21493	8633	262	33
7	5702	20673	8298	259	33
8	5278	21437	8714	247	32
9	5292	21147	8419	248	28
10	6168	21172	8946	239	51
Trung bình	5815.5	20835.2	8471.8	224.8	33.6

2. Biểu đồ (cột) thời gian thực hiện



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



## II. Kết luận:

Các thuật sort ở trên em đều viết bằng ngôn ngữ Python. Python chạy rất chậm so với Cpp. Lý do theo em tìm hiểu được là vì Python được thực thi bởi trình thông dịch, chạy từng dòng một nên tốc độ sẽ chậm hơn rất nhiều so với Cpp.

Vì vậy em tách ra làm 2 sơ đồ, 1 sơ đồ so sánh 3 thuật Heap Sort, Merge Sort và Quick Sort chạy trong ngôn ngữ Python và một sơ đồ riêng để so sánh 2 hàm sort trong Python và Cpp.

Nhận xét:

- Thuật chạy chậm nhất là Heap Sort (mặc dù độ phức tạp tính toán trên lý thuyết là nhỏ nhất  $O(n \log n)$ ), với thời gian chạy trung bình là 20835.2 ms ~ 20s. Tiếp đến là Merge Sort với trung bình là 8471.8 ms ~ 8s và nhanh nhất trong 3 thuật là Quick Sort với thời gian chạy trung bình 5815.5 ms ~ 5s.
- Trong quá trình chạy lần đầu tiên thì Merge Sort chạy rất chậm. Sau đó em có thay đổi lại cách chọn Pivot bằng cách lấy trung vị của 3 phần tử đầu, giữa và cuối thì thời gian chạy nhanh hơn khá nhiều.
- `std::sort()` trong C++ và `numpy.sort()` trong Python có thời gian chạy vượt trội so với các thuật trên.
- Thư viện numpy có thêm các build-in functions cho ta tùy chọn thuật toán sắp xếp cho hàm sort với cú pháp `numpy.sort(arr, kind="alg")`. Em có chạy thử các hàm này thì thời gian chạy khá nhanh.
- Thông thường `std::sort()` và `numpy.sort()` có thời gian chạy ngang nhau nhưng ta có thể tối ưu thời gian chạy của `numpy.sort()` bằng cách load data từ các file npy. Đó là lý do trong sơ đồ trên thì `numpy.sort()` chạy nhanh hơn `std::sort()` vì ban đầu em khởi tạo các file npy để xử lý dữ liệu. C++ không thể đọc trực tiếp data từ file npy nên em chuyển các file npy về file txt để chạy thuật (code cụ thể trong file `convert.py`).

### III. *Thông tin chi tiết – link github*

Link Repo: <https://github.com/dvckl3/IT003.P21.CTTN/tree/main/DSA%20Sorting>