

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 25/02 – 04/03

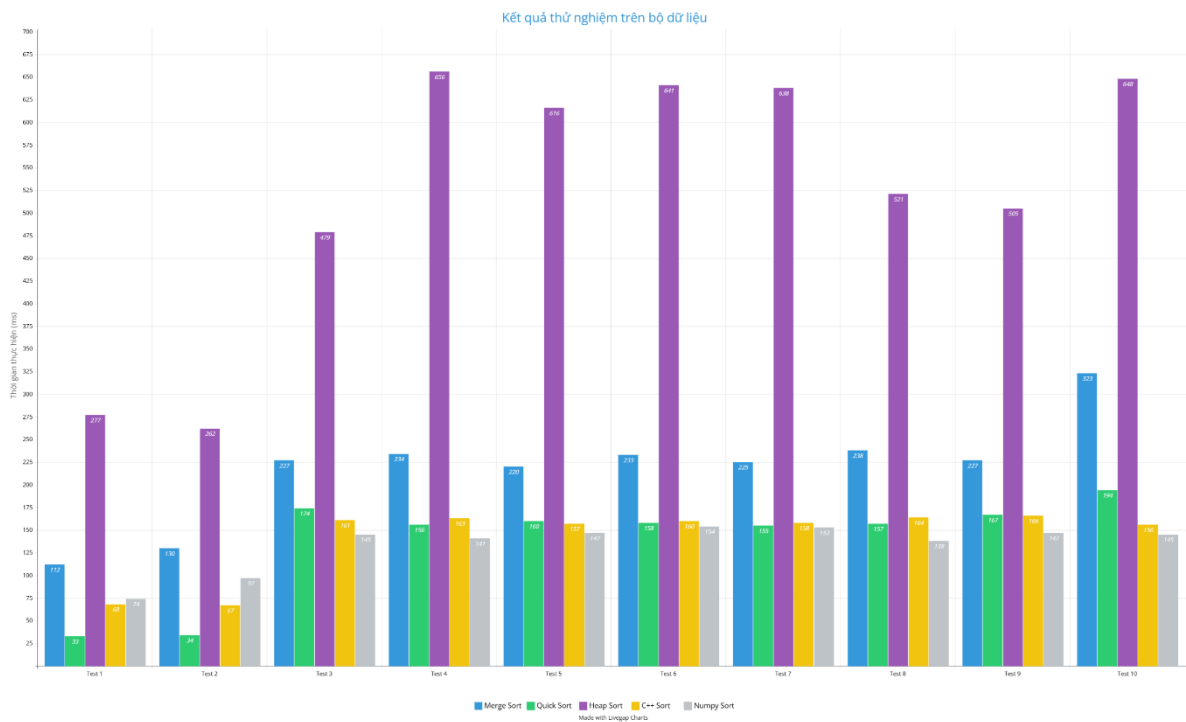
Sinh viên thực hiện: Nguyễn Trọng Nhân – 24520023

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 hiện¹

Dữ liệu	Thời gian thực hiện (ms)				
	Quicksort	Heapsort	Mergesort	sort (C++)	sort (numpy)
1	33	277	112	68	74
2	34	262	130	67	97
3	174	479	227	161	145
4	156	656	234	163	141
5	160	616	220	157	147
6	158	641	233	160	154
7	155	638	225	158	153
8	157	521	238	164	138
9	167	505	227	166	147
10	194	648	323	156	145
Trung bình	138.8	524.3	216.9	142.0	134.1

2. Biểu đồ (cột) thời gian thực hiện



¹ Số liệu chỉ mang tính minh họa

II. Kết luận:

- Kết quả cho thấy dùng Numpy sort là nhanh nhất với thời gian thực thi trung bình 134.1 ms, nhanh thứ nhì là Quicksort với tốc độ trung bình 138,8ms, nhanh thứ ba là hàm Sort có sẵn của C++ với thời gian trung bình là 142 ms, nhanh thứ tư là Merge Sort với thời gian thực thi trung bình là 216,9 ms và chậm nhất là Heap Sort với thời gian là 524,3 ms. Điều này giải thích được lí do tại sao Heap Sort không phổ biến trong các thuật toán sắp xếp, mặc cho độ phức tạp lý thuyết là $O(n \log n)$.
- Theo quan sát biểu đồ ta còn thấy được Numpy Sort rất ổn định trong đa số trường hợp, hàm Sort của C++ tuy chỉ chậm hơn một chút nhưng vẫn giữ tính ổn định cao. Quicksort có vẻ tỏ ra vượt trội đối với mảng sắp xếp tăng dần và giảm dần.
- Với thời gian chạy ổn định trong đa số trường hợp, hàm Sort của thư viện C++ chứng minh được lí do tại sao nó được sử dụng phổ biến nhất. Tuy Numpy chạy nhanh nhưng nếu ở một chương trình phức tạp có nhiều truy vấn khác nhau thì ngôn ngữ C/C++ sẽ tỏ ra được ưu thế vượt trội về mặt hiệu suất.

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

<https://github.com/sup3rshy/IT003.P21.CTTN/tree/main/Sorting%20Report>