Lớp: IT003.Q21.TTNT

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Sinh viên thực hiện: Nguyễn Khánh Vân

Mã số sinh viên: 25522051

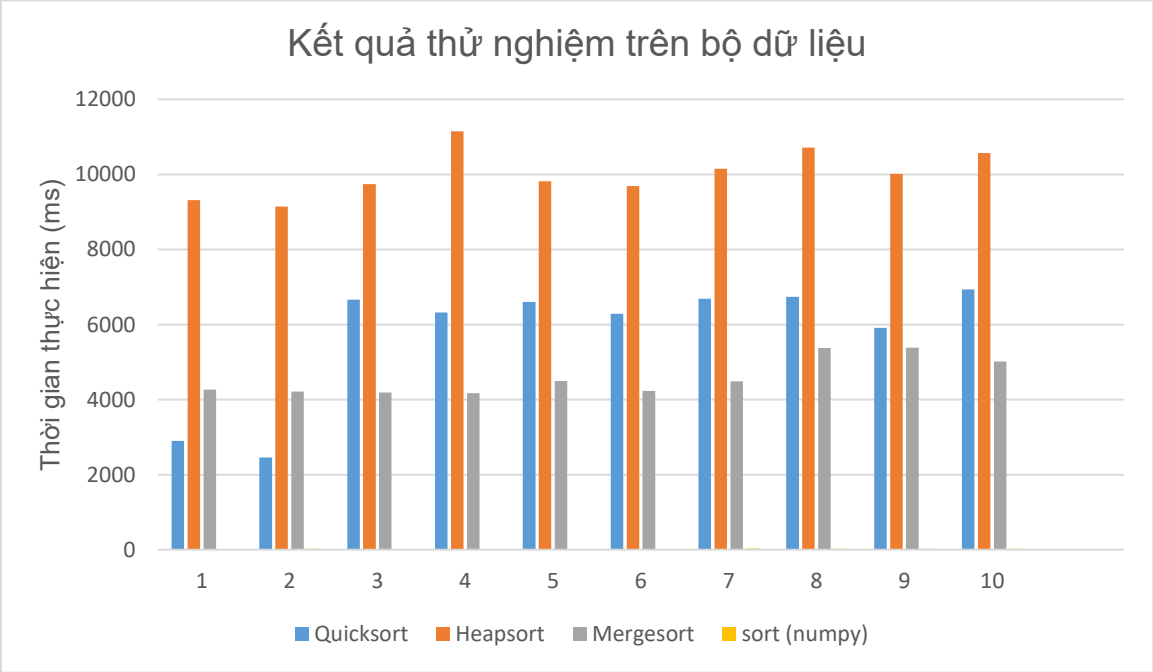
Nội dung báo cáo:

I. Kết quả thử nghiệm

1. Bảng thời gian thực hiện

Dữ liệu	Thời gian thực hiện (ms)			
	Quicksort	Heapsort	Mergesort	sort (numpy)
1	2900	9314	4265	10
2	2460	9147	4213	29
3	6660	9745	4187	10
4	6324	11147	4172	12
5	6601	9815	4495	15
6	6284	9688	4235	17
7	6687	10149	4485	32
8	6740	10711	5373	31
9	5910	10019	5384	30
10	6934	10571	5020	25
Trung bình	5750	10031	4583	21

2. Biểu đồ (cột) thời gian thực hiện



II. Kết luận:

- **Quicksort:** Thuật toán phụ thuộc mạnh vào trạng thái của dữ liệu, không ổn định và phụ thuộc vào cách chọn phần tử chốt. Thuật toán chạy nhanh ở dãy

được sắp xếp sẵn (dãy 1 và 2) (2460ms - 2900ms) và tăng thời gian thực thi khi sắp xếp dãy ngẫu nhiên (dãy 3 đến 10) (~6000 ms - 6900 ms). Chọn pivot ở chính giữa dãy có thể tránh một vài trường hợp đặc biệt, gây tăng thời gian thực thi.

- **Heapsort:** Thuật toán chạy chậm nhất trong mọi trường hợp, thời gian thực thi trung bình ~10031ms. Khi sử dụng ngôn ngữ Python, việc gọi đệ quy hàm c1 (hàm sắp) và thực hiện phép hoán vị liên tục tạo ra gánh nặng cho trình thông dịch Python khiến thuật toán chạy chậm hơn.
- **Mergesort:** Thuật toán có tính ổn định cao nhất trong số 3 thuật toán tự viết. Thời gian chạy luôn dao động trong khoảng 4200ms - 5300ms bất kể thứ tự của dữ liệu đầu vào.
- **sort (numpy):** Thuật toán chạy nhanh nhất trong tất cả thuật toán, mất trung bình ~21ms với mỗi mẫu dữ liệu, nhanh gấp 200 - 500 lần so với các thuật toán khác. Do hàm sort của numpy được viết và biên dịch bằng ngôn ngữ C/C++, loại bỏ được độ trễ của trình thông dịch Python nên nhanh hơn các thuật toán tự viết còn lại.

III. Thông tin chi tiết

Link Github: https://github.com/khanhvanng/IT003.Q21.TTNT-Sorting-Algorithm