Lớp: IT003.Q21.TTNT

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Sinh viên thực hiện: Đỗ Hải Yến

Mã số sinh viên: 25522133

Nội dung báo cáo:

1. Mục đích thử nghiệm

Bài thử nghiệm nhằm đánh giá, so sánh thời gian thực hiện các thuật toán so sánh khác nhau bằng ngôn ngữ python. Qua đó, làm rõ nguyên lý hoạt động của những thuật toán sắp xếp cơ bản, quan sát sự ảnh hưởng của dữ liệu đầu vào đến hiệu năng của từng bài toán.

2. Môi trường và dữ liệu

Do thời gian thực hiện thuật toán phụ thuộc lớn vào sức mạnh xử lý CPU và tốc độ bộ nhớ nên kết quả quá trình thử nghiệm cũng khác nhau.

Cấu hình phần cứng:

- Thiết bị: Laptop Nitro ANV15-51
- Vi xử lý (CPU): Intel Core i5-13420H
- Bộ nhớ trong (RAM): 16GB
- Hệ điều hành: Window 11 64-bit

Ngôn ngữ và thư viện: Python 3, NumPy, Pandas, Matplotlib.

Bộ dữ liệu: gồm 10 tập, mỗi tập chứa 1.000.000 phần tử

3. Cơ sở lý thuyết

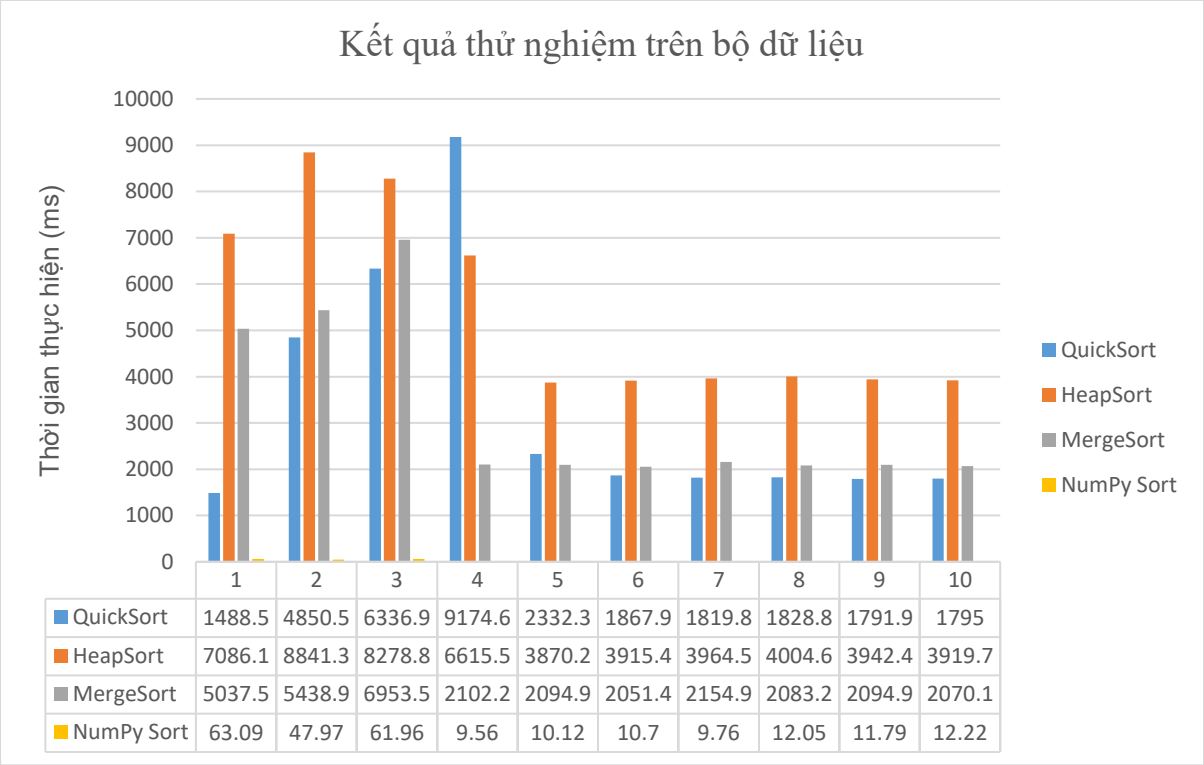
- Quick sort: phương pháp này chọn một phần tử làm chốt, phân chia mảng thành 2 phần nhỏ hơn và lớn hơn chốt, sau đó đệ quy cho đến khi mảng con còn 1 hoặc 0 phần tử. Độ phức tạp thời gian trung bình là O(N log N), trường hợp tệ nhất là O(N^2).
- Heap sort: phương pháp này thực hiện dựa trên một cây nhị phân hoàn chỉnh. Nguyên lý là xây dựng thuật toán tìm phần tử lớn nhất làm đỉnh, sau đó hoán đổi phần đỉnh này dần về cuối mảng chưa sắp xếp và lặp lại cho phần còn lại. Độ phức tạp thời gian luôn ở mức O(N log N).
- Merge sort: phương pháp này chia đôi mảng liên tục cho đến khi mảng con chỉ còn lại một phần tử. Sau đó lần lượt ghép các mảng con này lại với nhau theo thứ tự tăng dần để tạo thành mảng hoàn chỉnh. Độ phức tạp thời gian luôn là O(N log N).
- NumPy sort: được cài đặt bằng ngôn ngữ C ở tầng thấp, kết hợp linh hoạt nhiều thuật toán khác nhau để đạt được hiệu suất cao nhất.

I. *Kết quả thử nghiệm*

1. *Bảng thời gian thực hiện*

Dữ liệu	Thời gian thực hiện (ms)			
	Quicksort	Heapsort	Mergesort	sort (NumPy)
1	1488.48	7086.13	5037.52	63.09
2	4850.54	8841.29	5438.9	47.97
3	6336.9	8278.8	6953.49	61.96
4	9174.56	6615.47	2102.22	9.56
5	2332.34	3870.17	2094.94	10.12
6	1867.92	3915.4	2051.38	10.7
7	1819.75	3964.47	2154.94	9.76
8	1828.79	4004.61	2083.17	12.05
9	1791.92	3942.38	2094.88	11.79
10	1795.02	3919.72	2070.07	12.22
Trung bình	3328.62	5443.84	3208.15	24.92

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



II. *Kết luận:*

Qua quá trình nghiên cứu và thử nghiệm trên 1.000.000 phần tử, kết quả cho thấy các thuật toán sắp xếp đem lại hiệu năng khác nhau. Trong số các thuật toán được thử nghiệm, NumPy sort là thể hiện sức mạnh vượt trội với thời gian thực hiện nhanh nhất (9ms - 65ms) nhờ việc vận dụng linh hoạt các thuật toán khác nhau, tối ưu bộ nhớ đệm hiệu quả. Ngược lại, Heap Sort bằng python thuần lại chạy chậm nhất (3800ms – 8800ms) do đặc thù nhảy bước của cấu trúc cây gây trượt bộ nhớ đệm liên tục. Merge sort cho thấy hiệu năng khá ổn định (duy trì trong mức 2000ms ở mảng số nguyên). Quicksort biến động khá mạnh, tùy thuộc vào trạng thái phân bố của dữ liệu (1400ms – 9100ms). Kết quả thực nghiệm khẳng định rằng độ phức tạp Big O chỉ mang tính lý thuyết, trong thực tế, việc sử

dụng các thư viện chuyên dụng như NumPy là tiêu chuẩn bắt buộc để tối ưu hóa hiệu năng.

III. Thông tin chi tiết:

Link sản phẩm: https://github.com/dohaiyen67494/Sorting_Algorithms