

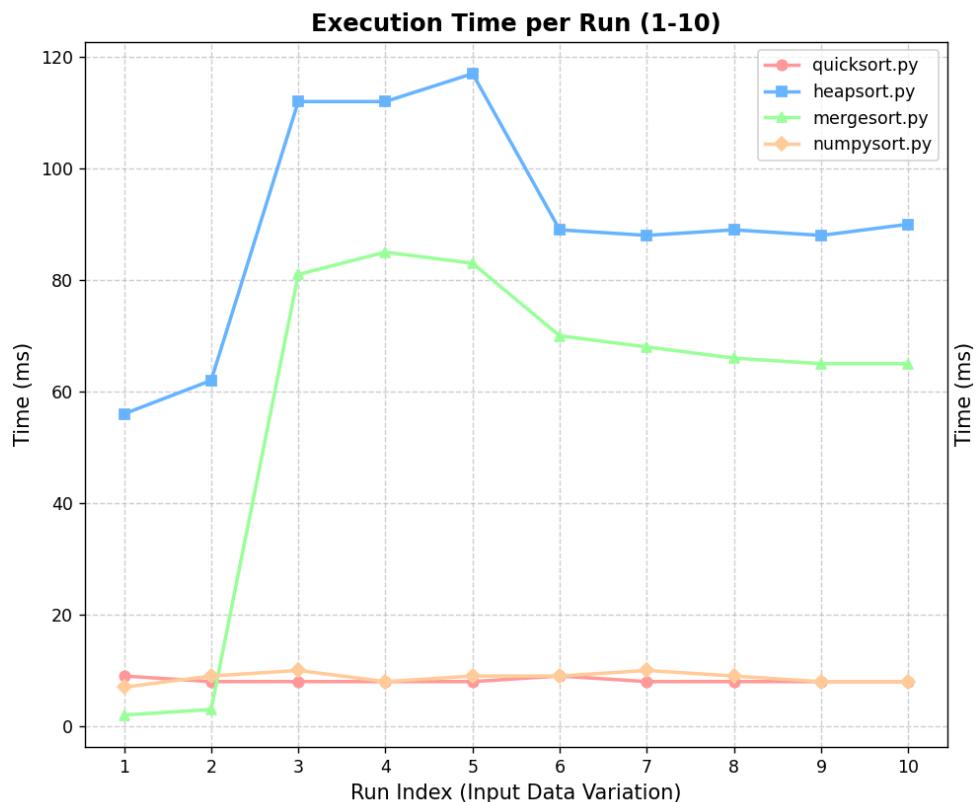
BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Sinh viên thực hiện: Lê Phú Trọng (25521937)

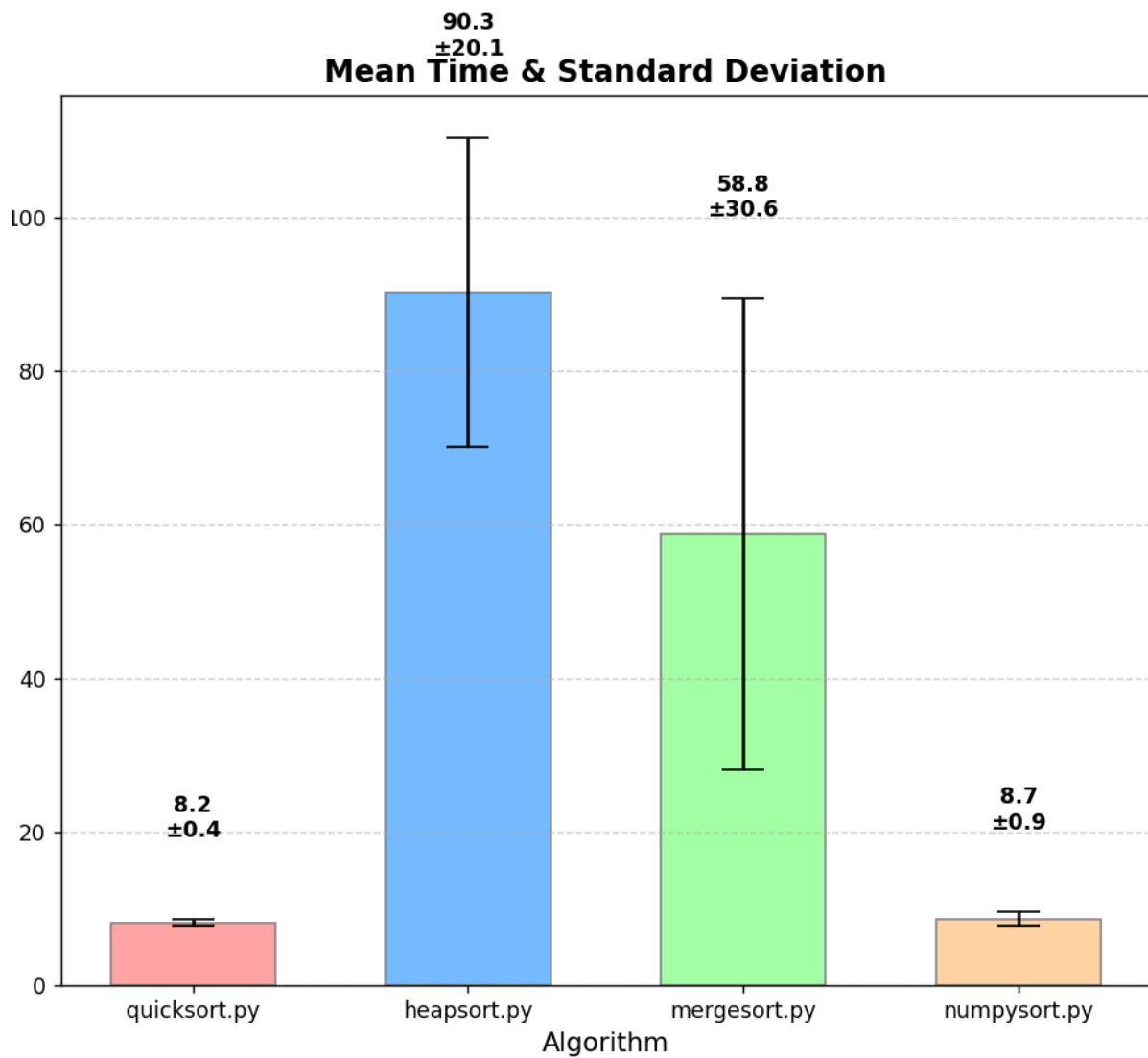
Nội dung báo cáo: Thực nghiệm các giải thuật sắp xếp nội

I. Kết quả thử nghiệm**1. Bảng thời gian thực hiện**

Dữ liệu	Thời gian thực hiện (ms)			
	Quicksort	Heapsort	Mergesort	sort (numpy)
1	9	56	2	7
2	8	62	3	9
3	8	112	81	10
4	8	112	85	8
5	8	117	83	9
6	9	89	70	9
7	8	88	68	10
8	8	89	66	9
9	8	88	65	8
10	8	90	65	8
Trung bình	8.2	90.3	58.8	8.7
Độ lệch chuẩn	0.42	20.1	30.6	0.95

2. Biểu đồ thời gian thực hiện

Hình 1: Biểu đồ đường thẳng thời gian chạy các thuật toán qua 10 tập dữ liệu



Hình 2: Biểu đồ cột thời gian trung bình và độ lệch chuẩn của từng loại thuật toán

II. Kết luận

Dựa trên thời gian chạy trung bình, thứ hạng các thuật toán lần lượt là: Quicksort, Numpysort, Mergesort, Heapsort.

Quicksort dù thường được biết đến là một thuật toán bất ổn định khi trường hợp tệ nhất độ phức tạp có thể đạt đến $O(N^2)$. Tuy nhiên, khi chọn random pivot, trường hợp tệ nhất gần như không xảy ra.

Numpysort có tốc độ gần tương đương với Quicksort

Mergesort chậm hơn 2 thuật toán trên. Dù cùng có độ phức tạp là $O(N \log N)$, Mergesort vẫn cần thêm bộ nhớ phụ khi thực hiện copy mảng trong thao tác Merge.

Heapsort là thuật toán chậm nhất do thao tác Heapify truy cập bộ nhớ không liên tục, dẫn đến locality kém.

Tác động của dữ liệu cũng ảnh hưởng đến kết quả:

- Các mảng đã được sắp xếp thường cho thời gian chạy thấp hơn so với các mảng theo thứ tự ngẫu nhiên vì các thao tác so sánh và hoán đổi xảy ra với tần suất thấp.

- Sự khác biệt của kiểu dữ liệu số nguyên và số thực không quá lớn, nhưng vẫn tạo được biến động do đặc điểm xử lý số học phức tạp hơn của kiểu số thực.

Độ ổn định:

- Quicksort và Numpy sort có độ ổn định cao, độ lệch chuẩn nhỏ giữa các lần chạy.
- Mergesort và Heapsort có độ dao động lớn, cho thấy mức độ nhạy cảm với cache và trạng thái bộ nhớ của hệ thống.
- Nhìn chung, kết quả cho thấy sự nhất quán và phản ánh đúng hành vi thực tế của các thuật toán trong môi trường NumPy.

III. ***Thông tin chi tiết***

Thông tin chi tiết hơn về báo cáo, mã nguồn và dữ liệu thử nghiệm có thể được xem tại:
<https://github.com/lftroq/IT003.Q21.CTTN-W1-SortingReport>.