

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 25/02/2025 – 04/03/2025

Sinh viên thực hiện: Lê Phạm Khánh Linh

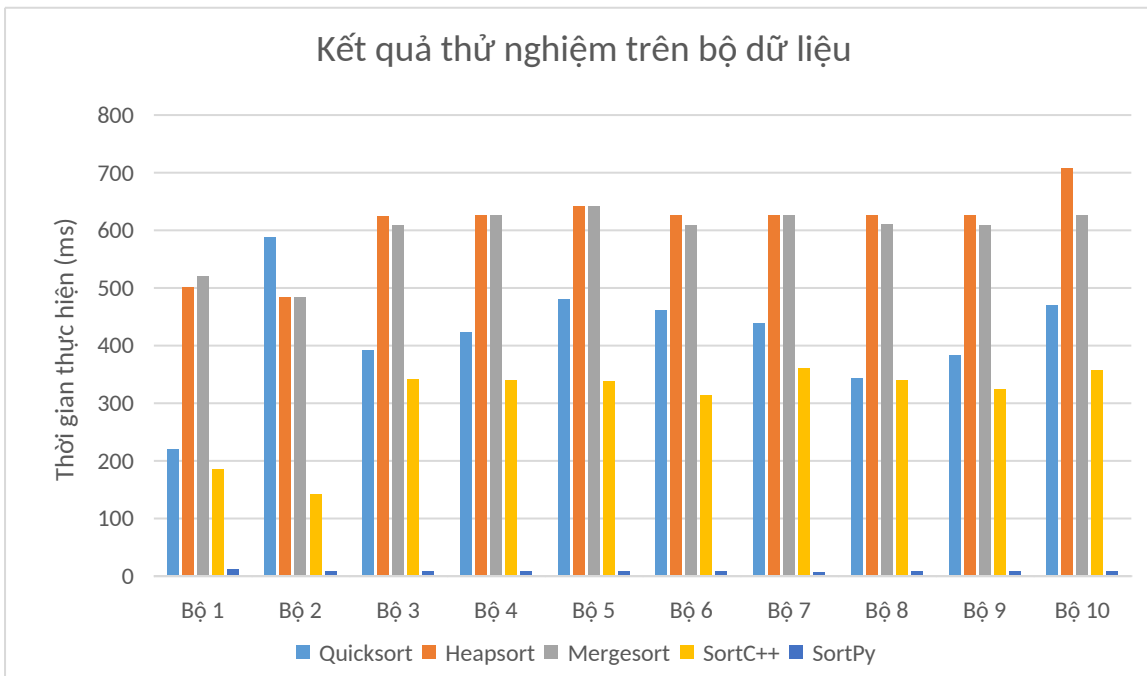
Nội dung báo cáo:

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 | 219.708 | 500.388 | 519.438 | 185.15 | 12.534 |
| 2 | 587.698 | 484.46 | 483.713 | 141.289 | 9.083 |
| 3 | 391.215 | 624.449 | 609.487 | 340.779 | 8.573 |
| 4 | 423.013 | 626.553 | 626.233 | 339.642 | 8.713 |
| 5 | 479.771 | 641.798 | 641.673 | 338.536 | 8.152 |
| 6 | 461.494 | 626.261 | 609.27 | 313.057 | 7.670 |
| 7 | 438.448 | 626.842 | 625.726 | 361.367 | 7.458 |
| 8 | 343.779 | 626.689 | 610.494 | 338.902 | 8.402 |
| 9 | 382.732 | 625.753 | 609.409 | 324.745 | 7.688 |
| 10 | 469.115 | 707.967 | 626.356 | 357.878 | 8.136 |
| Trung bình | 419.697 | 609.116 | 596.180 | 304.135 | 8.641 |

2. Biểu đồ (cột) thời gian thực hiện

¹ Số liệu chỉ mang tính minh họa

II. Kết luận:

- Sort (NumPy) là thuật toán nhanh nhất nhờ sử dụng Timsort kết hợp với vector hóa trong C, giúp tối ưu hóa hiệu suất với dữ liệu lớn.
- Sort (C++) tận dụng Introsort, tự động điều chỉnh thuật toán theo dữ liệu đầu vào, đảm bảo hiệu suất cao và ổn định, phù hợp cho hệ thống xử lý thời gian thực.
- Quicksort có tốc độ xử lý nhanh nhưng không ổn định do phụ thuộc vào cách chọn pivot, có thể làm tốc độ dao động đáng kể.
- Mergesort và Heapsort có hiệu suất gần tương đương. Mergesort yêu cầu bộ nhớ phụ, trong khi Heapsort xử lý cấu trúc heap, gây ảnh hưởng đến tốc độ.
 - Nếu cần tốc độ tối đa: NumPy là lựa chọn tốt nhất.
 - Nếu cần sự ổn định và đa năng: Sort C++ tối ưu hơn.
 - Nếu muốn tùy chỉnh thuật toán sắp xếp: Quicksort là lựa chọn linh hoạt.
 - Nếu không ưu tiên hiệu suất: Heapsort và Mergesort vẫn có thể dùng nhưng không phải phương án tối ưu.

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

- Link: https://github.com/lyns184/DSA_Sort.git