Lớp: IT003.P21.CTTN

**BÁO CÁO KẾT QUẢ THỬ NGHIỆM**

Thời gian thực hiện: 1/3/2025

**Sinh viên thực hiện: Đoàn Hồng Bảo**

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

1. ***Kết quả thử nghiệm***
   1. ***Bảng thời gian thực hiện[[1]](#footnote-1)***
   2. ***ms.41 ms.55 ms.71 ms.62 ms.96 ms.66 ms 13.53 ms 13.60 ms 13.72 ms***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dữ liệu** | **Thời gian thực hiện (ms)** | | | | |
| **Quicksort** | **Heapsort** | **Mergesort** | **sort (C++)** | **sort (numpy)** |
| 1 | 370 | 705 | 725 | 323 | 14 |
| 2 | 337 | 716 | 731 | 321 | 13 |
| 3 | 335 | 685 | 741 | 349 | 13 |
| 4 | 329 | 700 | 741 | 336 | 13 |
| 5 | 330 | 715 | 706 | 321 | 13 |
| 6 | 346 | 676 | 715 | 321 | 13 |
| 7 | 331 | 717 | 719 | 334 | 13 |
| 8 | 336 | 656 | 718 | 321 | 13 |
| 9 | 342 | 709 | 691 | 324 | 13 |
| 10 | 350 | 656 | 730 | 319 | 13 |
| Trung bình | 340.6 | 693.5 | 721.7 | 327.4 | 13.3 |

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

1. ***Kết luận:***

- sort (NumPy) là nhanh nhất với thời gian trung bình chỉ 13.3 ms, cho thấy tối ưu cực cao nhờ các thư viện cấp thấp và xử lý vector hoá.

- sort (C++) cũng rất nhanh với thời gian trung bình 327.4 ms, nhờ vào hiệu quả của thuật toán được chuẩn hoá trong thư viện STL.

- Quicksort có hiệu năng khá tốt với 340.6 ms nhưng vẫn chậm hơn sort (C++).

Heapsort và Mergesort đều chậm hơn đáng kể, với thời gian trung bình lần lượt là 693.5 ms và 721.7 ms. Điều này phù hợp với độ phức tạp thực tế của các thuật toán này, đặc biệt khi làm việc với dữ liệu lớn.

=> sort (NumPy) là lựa chọn tốt nhất nếu bạn đang làm việc với Python và muốn hiệu năng tối đa. Trong môi trường C++, std::sort vẫn là lựa chọn vượt ***trội về tốc độ.***

1. ***Thông tin chi tiết – link github, trong repo gibub cần có***
   1. Báo cáo
   2. Mã nguồn
   3. Dữ liệu thử nghiệm

1. [↑](#footnote-ref-1)