Lớp: IT003.P21.CTTN

**BÁO CÁO KẾT QUẢ THỬ NGHIỆM**

Thời gian thực hiện: 25/02/2025 – 26/02/2025

**Sinh viên thực hiện:** Bảo Quý Định Tân - 24520028

**Nội dung báo cáo:** Cho dữ liệu được tạo gồm 10 dòng, mỗi dòng chứa 1 triệu số:

* Dòng 1: Dữ liệu được sắp xếp tăng dần.
* Dòng 2: Dữ liệu được sắp xếp giảm dần.
* Dòng 3: Dữ liệu chỉ gồm các số nguyên từ khoảng -10^9 đến 10^9.
* Dòng 4: Dữ liệu chỉ gồm các số nguyên từ khoảng -10^18 đến 10^18.
* Dòng 5: Dữ liệu chỉ gồm các số thực từ khoảng -10^18 đến 10^18.
* Dòng 6: Dữ liệu chỉ gồm các số nguyên vừa từ khoảng 0 đến 10^6.
* Dòng 7: Dữ liệu chỉ gồm các số nguyên nhỏ từ khoảng 0 đến 10^3.
* Dòng 8: Dữ liệu chỉ gồm các số thực vừa từ khoảng -10^6 đến 10^6.
* Dòng 9 và 10: Dữ liệu trộn ngẫu nhiên số nguyên và số thực.

Thực hiện chạy thử các phương pháp sắp xếp gồm: QuickSort, HeapSort, MergeSort, sort (c++), sort (numpy) để so sánh thời gian thực thi và đánh giá kết quả. Mỗi phương pháp sắp xếp được chạy 10 lần và tính trung bình thời gian thực thi trên mỗi dòng dữ liệu.

*Hàm QuickSort được thay đổi trong khâu cài đặt, thay vì chọn 1 phần tử làm pivot, ta chọn 3 phần tử ở đầu, giữa, đuôi để xem xét làm pivot, điều này giúp tránh rơi vào trường hợp pivot tệ làm cho thuật toán chạy chậm.*

1. ***Kết quả thử nghiệm***
   1. ***Bảng thời gian thực hiện[[1]](#footnote-1)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dữ liệu** | **Thời gian thực hiện (ms)** | | | | |
| **Quicksort** | **Heapsort** | **Mergesort** | **sort (C++)** | **sort (numpy)** |
| 1 | 16.3888 | 81.090 | 183.972 | 17.089 | 13.899 |
| 2 | 37.2482 | 87.258 | 193.994 | 13.863 | 13.238 |
| 3 | 80.7479 | 274.448 | 272.749 | 75.776 | 14.065 |
| 4 | 80.3387 | 238.588 | 278.33 | 77.585 | 13.294 |
| 5 | 84.0643 | 246.903 | 273.913 | 75.117 | 13.505 |
| 6 | 83.7631 | 232.831 | 280.995 | 74.181 | 12.335 |
| 7 | 312.89 | 187.729 | 238.87 | 47.041 | 7.074 |
| 8 | 81.8626 | 206.155 | 274.505 | 74.423 | 13.479 |
| 9 | 81.5749 | 198.284 | 278.232 | 75.663 | 13.307 |
| 10 | 79.1131 | 205.817 | 269.02 | 75.625 | 13.743 |
| Trung bình | 93.8 | 195.9103 | 254.458 | 60.6363 | 12.794 |

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

A graph of different colored columns

AI-generated content may be incorrect.

A graph of different colored squares

AI-generated content may be incorrect.

1. ***Kết luận:***

* Hàm sort của numpy đã được tối ưu bằng nhiều cách thức khác ngoài thuật toán sắp xếp thông thường nên có thời gian chạy nhanh hơn hẳn những phương pháp sắp xếp còn lại.
* Hàm sort có sẵn của c++ cũng được tối ưu bằng cách kết hợp nhiều phương pháp sắp xếp lại với nhau và một số kỹ thuật khác nên có thời gian chạy nhanh hơn hẳn những phương pháp sắp xếp khác, thua mỗi hàm sort của numpy.
* QuickSort có thời gian tương đối nhanh so với HeapSort và MergeSort bởi cách chọn pivot được nêu trên, tuy nhiên trong trường hợp tệ, QuickSort vẫn cho thời gian chạy khá chậm (có thể thấy ở test 7).
* HeapSort và MergeSort có thời gian thực thi tương đương nhau, nhưng HeapSort trông nhỉnh hơn đôi chút, nhất là khi mảng được sắp xếp sẵn tăng dần và giảm dần bởi MergeSort trong mọi trường hợp đều thực hiện đầy đủ toàn bộ các bước, nhưng HeapSort có thể dừng lại sớm hơn, tránh thực hiện những bước thừa và việc xây dựng lên Heap trong HeapSort có thể tận dụng tốt tính chất mảng đã được sắp xếp sẵn.

1. ***Thông tin chi tiết :*** [**BoquDnhTan/Compare-Sorting-Methods: DSA homework 2**](https://github.com/BoquDnhTan/Compare-Sorting-Methods)
   1. **Báo cáo:** Sorting Report - 24520028.docx
   2. **Mã nguồn:** Code trong các folder cụ thể: Sort methods, Visualization.
   3. **Dữ liệu thử nghiệm:** input.inp được tạo ra từ gen\_input.cpp

1. Số liệu chỉ mang tính minh họa [↑](#footnote-ref-1)