

BÁO CÁO THỰC NGHIỆM CÁC THUẬT TOÁN SẮP XẾP

---- Môn Cấu trúc dữ liệu và giải thuật - IT003.N21.CTTN ----

Báo cáo viên: Châu Thế Vĩ

MSSV: 22521653

Lớp: KHTN2022

Email: 22521653@gm.uit.edu.vn

Giới thiệu

Báo cáo về thời gian thực thi của các thuật toán sắp xếp như Quicksort, Heapsort, Mergesort và hàm `std::sort()` trong thư viện chuẩn của C++.

Chuẩn bị

- Bộ dữ liệu kiểm thử gồm 10 dãy số thực, trong đó có 1 dãy số tăng dần, 1 dãy số giảm dần và 8 dãy số ngẫu nhiên.
 - “maketest.cpp” là file code dùng để tạo các bộ test.
 - “test1.txt” là bộ test có thứ tự tăng dần.
 - “test2.txt” là bộ test có thứ tự giảm dần.
 - Các file từ “test3.txt” đến “test10.txt” là các bộ test được tạo ngẫu nhiên.
- Code cài đặt: Quicksort, Heapsort, Mergesort được lưu tương ứng trong “QuickSort.cpp”, “MergeSort.cpp”, “HeapSort.cpp”.
- Code sử dụng `std::sort` trong thư viện chuẩn của C++.
- Code thực hiện thuật toán và tính toán thời gian được lưu trong file code “main.cpp”.
- Biểu đồ:
 - Được vẽ bằng Python.
 - File code: “createChart.py” và file ảnh: “Chart.png”.

Tất cả file liên quan đều được lưu trữ trong [Github](https://github.com/chauthevi2004) (<https://github.com/chauthevi2004>)

KẾT QUẢ THỰC NGHIỆM

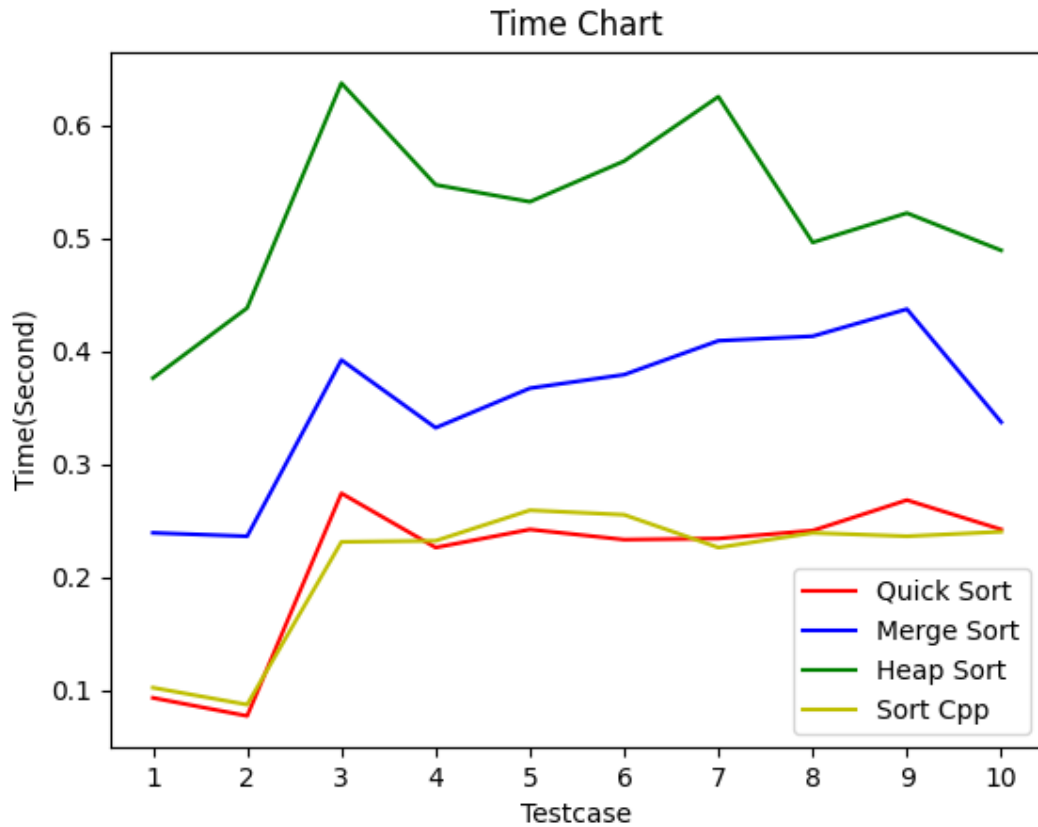
(được chạy trên máy tính cá nhân có cấu hình:

11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz).

Bảng dữ liệu thời gian thực thi các thuật toán sắp xếp (đơn vị giây):

Testcase	Quicksort	Heapsort	Mergesort	Std::sort
1	0.093	0.376	0.239	0.102
2	0.077	0.438	0.236	0.087
3	0.274	0.637	0.392	0.231
4	0.226	0.547	0.332	0.232
5	0.242	0.532	0.367	0.259
6	0.233	0.568	0.379	0.255
7	0.234	0.625	0.409	0.226
8	0.241	0.496	0.413	0.239
9	0.268	0.522	0.437	0.236
10	0.242	0.489	0.337	0.24

Biểu đồ dữ liệu



Nhận xét:

- Các thuật toán có độ ổn định cao và thời gian thực thi khác nhau trên các bộ test khác nhau. Ví dụ, trong trường hợp các bộ test đã được sắp xếp, các thuật toán sẽ có thời gian thực thi nhanh hơn.
- Thuật toán Quicksort và hàm `std::sort` có thời gian thực thi tương đương nhau, tuy nhiên, hàm `std::sort` của C++ có tính ổn định và thời gian thực thi tốt hơn một chút so với Quicksort.
- Mergesort là một thuật toán có thời gian thực thi chậm hơn Quicksort và `std::sort` nhưng lại nhanh hơn Heapsort. Tuy nhiên, việc sử dụng nhiều bộ nhớ để tạo ra các mảng trung gian là một nhược điểm của thuật toán này.
- HeapSort là thuật toán có thời gian thực thi chậm hơn các thuật toán còn lại và độ ổn định của nó không quá cao.