

# BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 11/03 – 16/03/2022

**Sinh viên thực hiện: Lê Hoài Thương**

## Nội dung báo cáo:

Chạy thử nghiệm các chương trình sắp xếp trên các bộ dữ liệu đầu vào ngẫu nhiên. Ghi lại thời gian thực thi của từng chương trình trên từng tests, lập bảng kết quả, vẽ biểu đồ kết quả và nhận xét. Cụ thể:

- Đối tượng thử nghiệm: chương trình sắp xếp được cài đặt theo thuật toán QuickSort, HeapSort, MergeSort và chương trình gọi hàm `std::sort` (C++ STL).

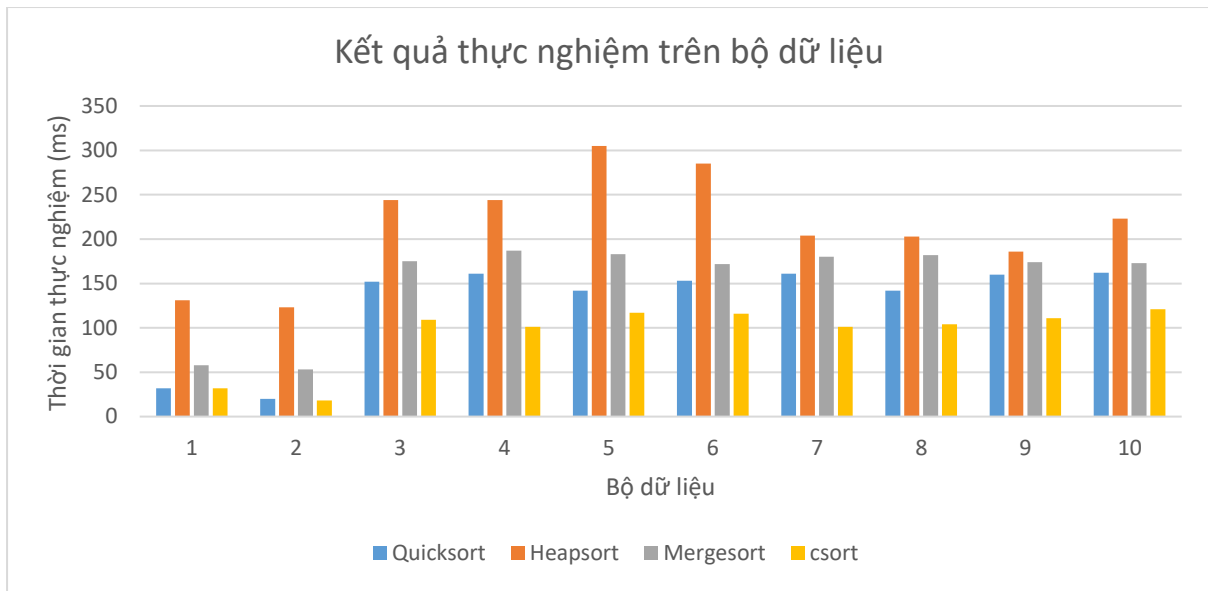
– Dữ liệu thử nghiệm: gồm 10 tests đầu vào, mỗi test bao gồm 1 triệu số thực trong đoạn  $[-10^9, 10^9]$  được sinh ngẫu nhiên. Trong đó, test 1 được sắp xếp tăng dần, test 2 được sắp xếp giảm dần, 8 tests còn lại có thứ tự ngẫu nhiên.

## I. Kết quả thử nghiệm

### 1. Bảng thời gian thực hiện

Dữ liệu	Thời gian thực hiện (ms)			
	Quicksort	Heapsort	Mergesort	sort (C++)
1	32	131	58	32
2	20	123	53	18
3	152	244	175	109
4	161	244	187	101
5	142	305	183	117
6	153	285	172	116
7	161	204	180	101
8	142	203	182	104
9	160	186	174	111
10	162	223	173	121
Trung bình	128.5	214.8	153.7	93

### 2. Biểu đồ (cột) thời gian thực hiện



## II. Kết luận:

- Nhìn chung, std Sort là thuật toán có thời gian thực thi trung bình nhỏ nhất, ngược lại, HeapSort có thời gian thực thi trung bình lớn nhất (lớn gấp 2 lần so với std Sort). - Trong các trường hợp tổng quát (chưa được sắp xếp sẵn), các chương trình đều có thời gian thực thi ổn định qua các tests. Cụ thể:
  - QuickSort và MergeSort có thời gian thực thi tương đương nhau. Đây là hai chương trình nhanh nhất.
  - Hàm std::sort có thời gian thực thi khá ngắn, chênh lệch không đáng kể so với MergeSort và QuickSort o HeapSort có thời gian thực thi lớn nhất (lớn gấp 2 lần so với hàm sort mặc định của C++).
- Khi thực thi trên hai tests đã được sắp xếp, cả bốn chương trình đều đạt được thời gian thực thi rất ngắn so với các trường hợp tổng quát, đặc biệt là QuickSort và std Sort (chỉ bằng 1/4 so với trường hợp tổng quát, và bằng 1/4 so với HeapSort và 1/2 MergeSort trong cùng trường hợp). Vậy: std Sort là hàm tối ưu thời gian chạy và thời gian cài đặt nhất, tiện lợi nhất .

## III. Thông tin chi tiết – link github, trong repo gibub cần có [Chirox03/IT003\\_homework \(github.com\)](https://github.com/Chirox03/IT003_homework)