

## BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Thời gian thực hiện: 01/03 – 16/03/2022

Sinh viên thực hiện: Hà Văn Hoàng – MSSV: 21520033

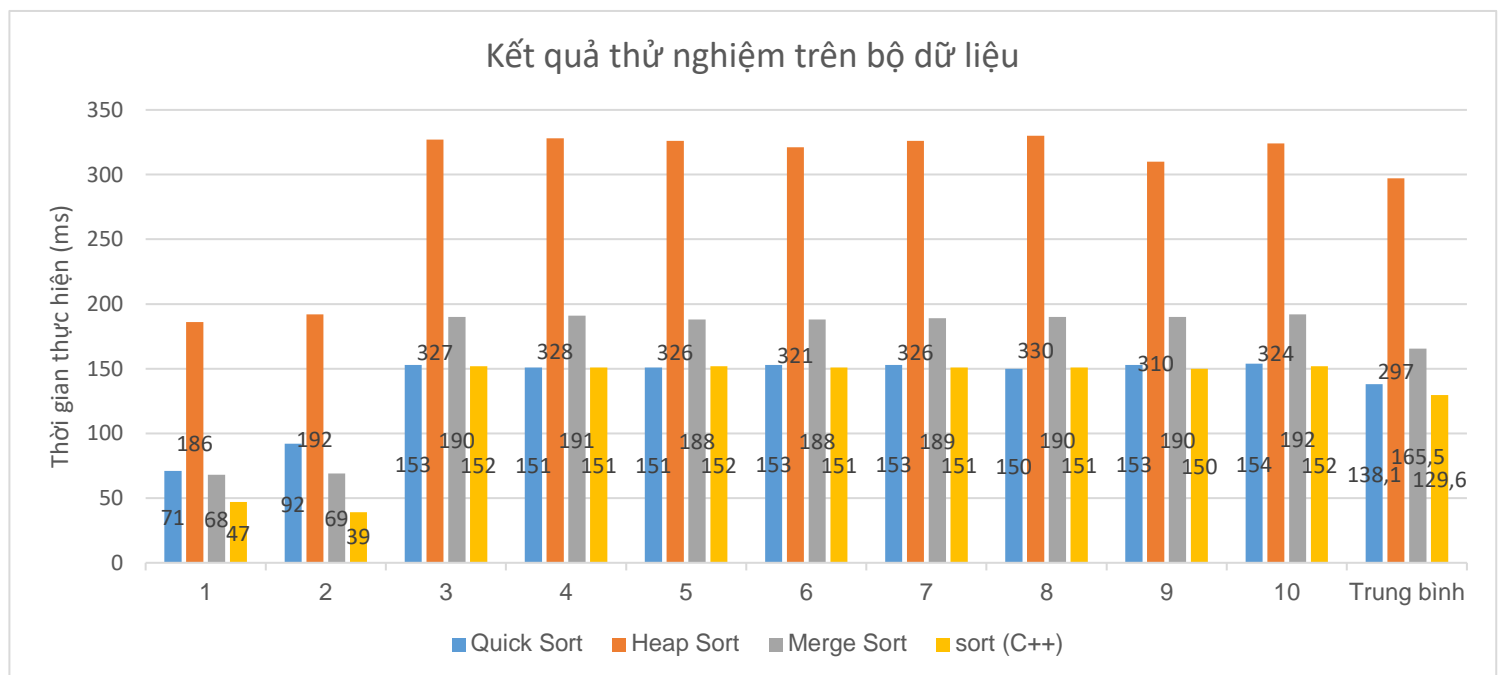
Nội dung báo cáo: Báo cáo kết quả thực nghiệm các thuật toán: Quicksort, Heapsort, Mergesort, và hàm sort (C++).

### I. Kết quả thử nghiệm

#### 1. Bảng thời gian thực hiện

Dữ liệu	Thời gian thực hiện (ms)			
	Quicksort	Heapsort	Mergesort	sort (C++)
1	71	186	68	47
2	92	192	69	39
3	153	327	190	152
4	151	328	191	151
5	151	326	188	152
6	153	321	188	151
7	153	326	189	151
8	150	330	190	151
9	153	310	190	150
10	154	324	192	152
Trung bình	138,1	297	165,5	129,6

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



### II. Kết luận:

- Nhìn chung: Quick Sort và sort (C++) sắp xếp với thời gian khá ngang nhau và sort (C++) có thời gian sắp xếp trung bình nhanh nhất. Còn Heap Sort thì sắp xếp với thời gian chậm nhất. Mảng đã sắp xếp sẵn luôn được sắp xếp nhanh hơn so với mảng ngẫu nhiên.
- Về thuật toán Quick Sort:  
Đây là một thuật toán được dùng phổ biến với khả năng sắp xếp rất nhanh (mặc dù trường hợp xấu nhất là  $O(N^2)$ ) và tiết kiệm bộ nhớ. Với mảng đã sắp xếp thì sắp xếp với thời gian  $< 0.1s$ , còn mảng ngẫu nhiên thì sắp xếp với thời gian khoảng  $0.15s$ .
- Về thuật toán Merge Sort:  
Đây là một thuật toán khá nhanh, nhưng buộc phải dùng thêm bộ nhớ. Với thời gian sắp xếp của mảng đã sắp xếp khoảng  $0.07s$  (nhanh hơn Quick Sort) và của mảng ngẫu nhiên khoảng  $0.19s$ .
- Về thuật toán Heap Sort:  
Đây là một thuật toán chậm nhất, nhưng bù lại không tốn thêm bộ nhớ, với độ phức tạp là  $O(N \log N)$ . Thời gian sắp xếp của mảng đã sắp xếp khoảng  $0.19s$  và của mảng ngẫu nhiên khoảng  $0.31$  tới  $0.33s$ .
- Về hàm `std::sort`:  
Đây là một hàm dùng một thuật toán gọi là IntroSort. Vì đây là hàm có sẵn, rất nhanh, rất thuận lợi nên được người lập trình sử dụng nhiều nhất. Thời gian sắp xếp của mảng đã sắp xếp khoảng  $0.04s$  và của mảng ngẫu nhiên khoảng  $0.15s$ .

### III. **Link github:** <https://github.com/hhoangcpascal/Sorting-Report>

- Bộ test 1 được sắp xếp tăng dần, bộ test 2 được sắp xếp giảm dần, 8 bộ test còn lại ngẫu nhiên. Tất cả các bộ test đều nằm trong thư mục `Test_Case`
- 4 bài code cho 4 thuật và hàm trên đều nằm trong thư mục `File_Cpp`