

# BÁO CÁO KẾT QUẢ THỬ NGHIỆM

Sinh viên thực hiện: Lại Đức Huy - 25520691

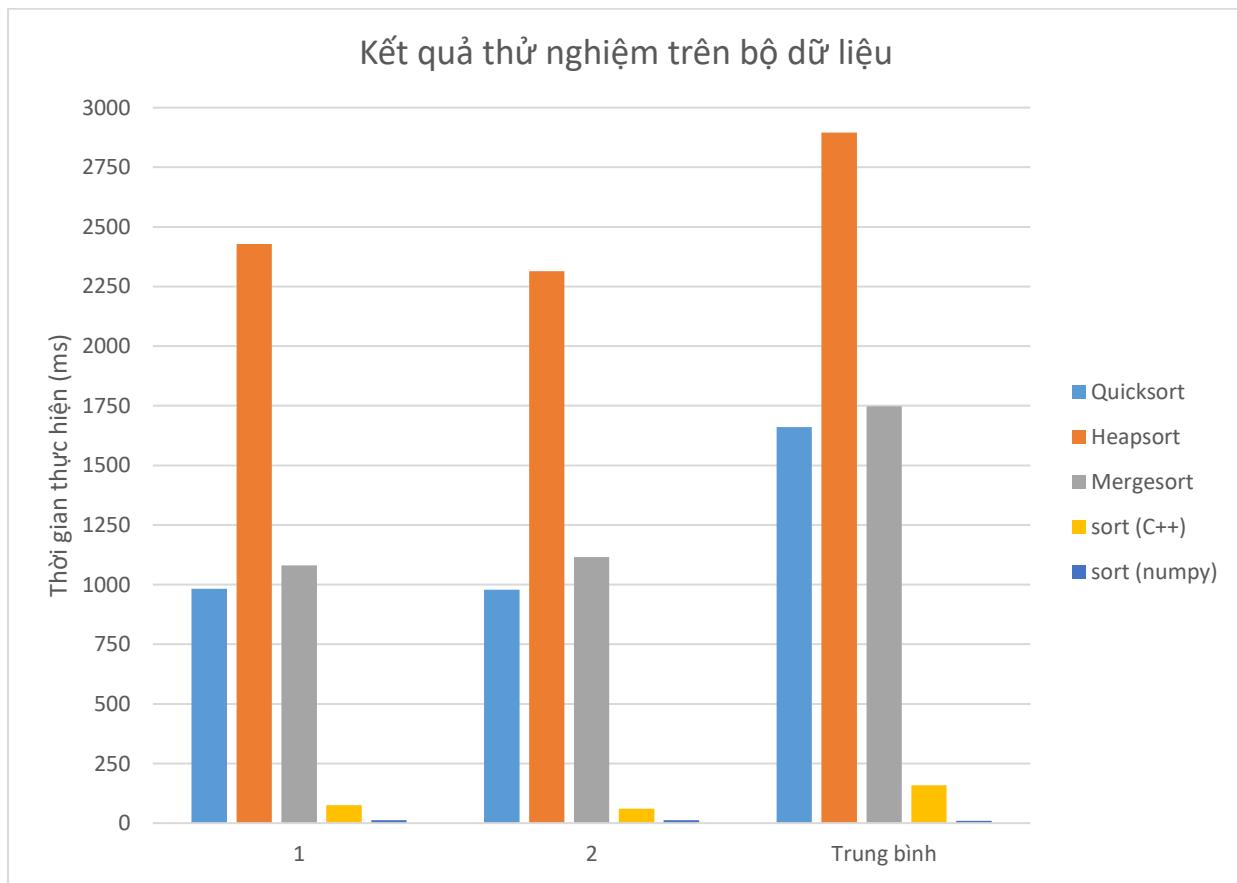
Nội dung báo cáo: Kết quả thử nghiệm các giải thuật sắp xếp nội

## I. Kết quả thử nghiệm

### 1. Bảng thời gian thực hiện<sup>1</sup>

Dữ liệu	Thời gian thực hiện (ms)				
	Quicksort	Heapsort	Mergesort	sort (C++)	sort (numpy)
1	983	2428	1081	75	12
2	979	2314	1115	61	12
3	1779	2925	1892	179	12
4	1697	2880	1912	180	12
5	1673	2858	1897	179	12
6	1861	3038	1920	180	8
7	2014	3050	1942	180	8
8	1873	3192	1920	181	8
9	1819	3147	1907	180	8
10	1920	3124	1894	180	8
Trung bình	1660	2896	1748	158	10

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



<sup>1</sup> Số liệu chỉ mang tính minh họa

## **II. Kết luận:**

Thuật sắp xếp nhanh nhất là numpy sort (trung bình khoảng 10ms), tiếp theo là sort của C++ (158ms), Quicksort (1660ms), Mergesort (1748ms) và chậm nhất là Heapsort (2896ms).

Numpy sort và C++ sort cho kết quả nhanh hơn vì được cài đặt trong thư viện của ngôn ngữ, sử dụng những kỹ thuật tối ưu phần cứng máy tính. Quicksort, Mergesort và Heapsort có thời gian chậm hơn nhiều do không có sự tối ưu phần cứng này. Heapsort chậm nhất do thực hiện nhiều phép hoán đổi phần tử xa nhau trong mảng, không tận dụng được cache máy tính nên chạy chậm hơn.

Về sự ổn định, numpy sort và C++ sort có độ dao động nhỏ. Cụ thể, numpy sort dao động từ 8ms đến 12ms (chênh nhau 4ms), C++ sort từ 61ms đến 180ms (chênh nhau 119ms, tuy nhiên phần lớn là khoảng 180ms). Sau đó là Mergesort (1080ms đến 1942ms, chênh nhau 861ms), Heapsort (2314ms và 3192ms, chênh 878ms) và ít ổn định nhất là Quicksort (979ms đến 2014ms, chênh 1035ms). Điều này cho thấy các thuật toán tự cài đặt có sự ổn định thấp hơn so với các hàm của thư viện.

Tóm lại, numpy sort là thuật toán nhanh nhất và cũng ổn định nhất, Heapsort có thời gian thực hiện chậm nhất và Quicksort là thuật toán kém ổn định nhất.

## **III. Thông tin chi tiết – link github, trong repo gibub cần có Github của báo cáo tại đây**

1. Báo cáo
2. Mã nguồn
3. Dữ liệu thử nghiệm