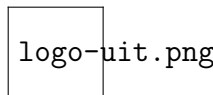


Đại học quốc gia TP.HCM
Trường đại học công nghệ thông tin



Ngành: Khoa học máy tính

Môn học: IT003.P21.CTTN

**BÁO CÁO KẾT QUẢ THỬ
NGHIỆM CÁC GIẢI THUẬT
SẮP XẾP**

Sinh viên:

Trần Quang Trường -
MSSV: 24521901

Giảng viên:

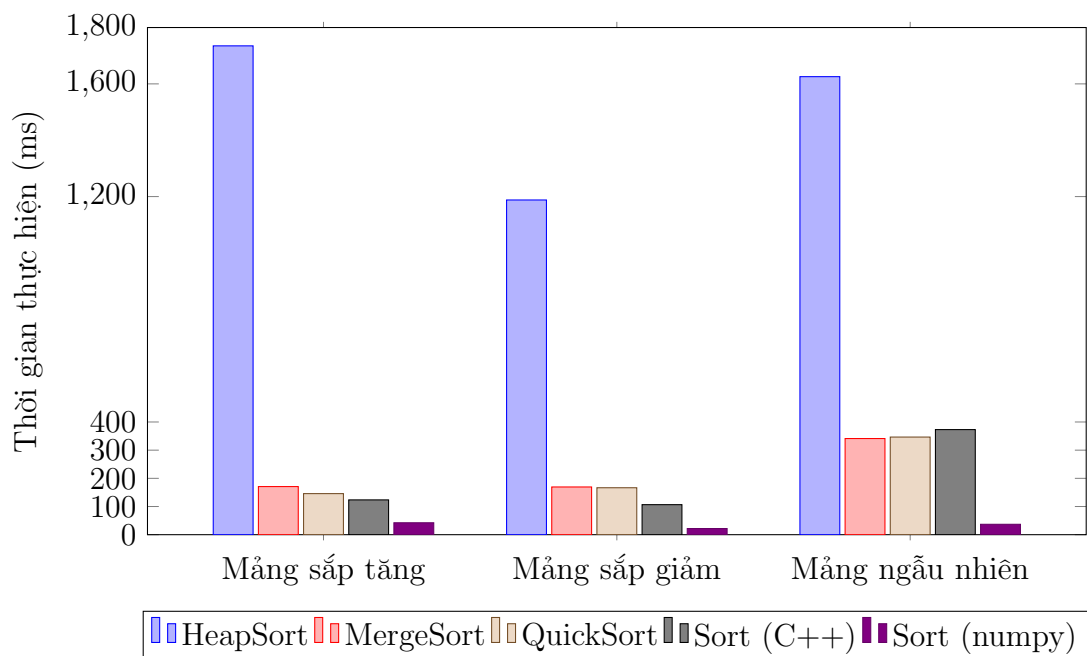
Nguyễn Thanh Sơn

1 Kết quả thực nghiệm

Bảng thời gian thực hiện

Dữ liệu	Thời gian thực hiện (ms)				
	Quicksort	Heapsort	Mergesort	sort (C++)	sort (numpy)
1	145.76	1734.94	170.75	123.47	42.29
2	166.61	1187.92	169.34	106.57	21.84
3	346.50	1625.60	341.23	372.98	36.88
4	357.15	1597.25	348.17	426.15	38.67
5	335.30	1545.19	348.76	362.23	27.57
6	347.36	1628.09	346.27	334.45	30.51
7	351.63	1626.49	340.45	338.88	29.88
8	349.29	1102.60	350.49	348.70	28.11
9	363.97	1615.25	348.99	395.35	33.95
10	363.10	1638.59	348.50	367.68	33.19
Trung bình	326.67	1520.89	340.19	317.74	32.79

Biểu đồ (cột) thời gian thực hiện



2 Kết luận

Từ kết quả thử nghiệm, có thể thấy rằng thuật toán **Sort (numpy)** hoạt động nhanh nhất, với thời gian trung bình chỉ khoảng 30.69ms. Ngược lại, **Heap Sort** là thuật toán chậm nhất, mất trung bình khoảng 1460.80ms để hoàn thành một lần sắp xếp.

Lý do **Sort (numpy)** nhanh như vậy là do nó sử dụng tối ưu hóa cấp thấp và xử lý dữ liệu theo kiểu vector hóa, giúp giảm đáng kể số lượng phép toán so với các thuật toán truyền thống. Trong khi đó, **Heap Sort** tuy có độ phức tạp $O(n \log n)$ nhưng lại thực hiện nhiều thao tác hoán đổi và truy cập bộ nhớ hơn, khiến nó kém hiệu quả hơn so với các thuật toán khác.

Trong nhóm các thuật toán truyền thống, **Merge Sort** và **Quick Sort** có tốc độ tương đối ổn, trung bình khoảng 296.50ms và 302.47ms. **Sort (C++)** (thuật toán sắp xếp mặc định của C++) cũng khá nhanh với trung bình 328.79ms.

Nhìn chung, nếu chỉ xét về tốc độ, **Sort (numpy)** là lựa chọn tốt nhất. Nếu không dùng thư viện bên ngoài, **Merge Sort** hoặc **Quick Sort** là hai lựa chọn hợp lý. Còn **Heap Sort**, trừ khi có lý do đặc biệt, không phải là phương án tối ưu.

3 Thông tin chi tiết

Mọi thông tin về mã nguồn, báo cáo và các tài liệu liên quan có thể được tìm thấy tại: <https://github.com/QioCas/Sorting-Algorithms-Report>