

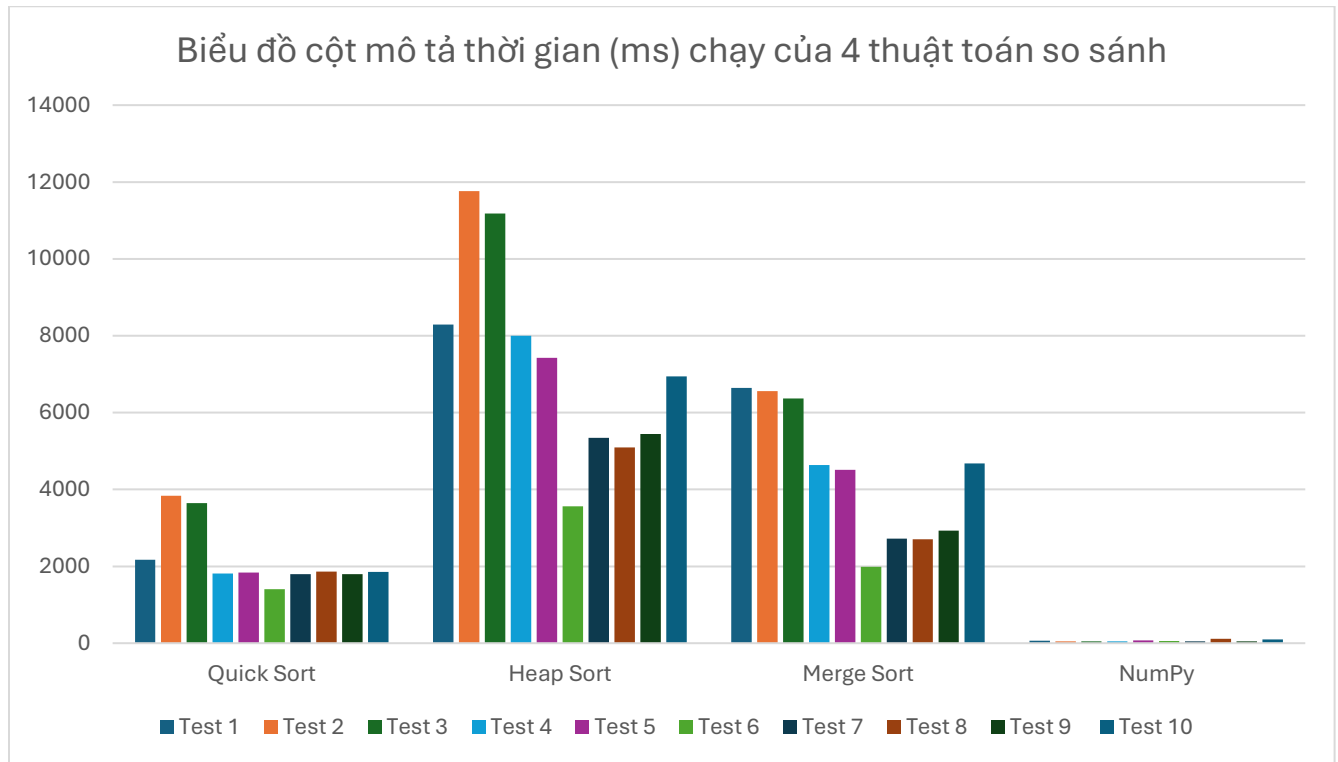
## Báo Cáo Chạy Thử Nghiệm Các Thuật Toán So Sánh

### 1. Bảng kết quả thử nghiệm:

	Quick Sort	Heap Sort	Merge Sort	NumPy
Test 1	2167	8291	6646	65
Test 2	3837	11759	6559	46
Test 3	3645	11180	6365	50
Test 4	1809	8000	4634	51
Test 5	1835	7428	4507	70
Test 6	1409	3564	1990	60
Test 7	1796	5343	2717	47
Test 8	1862	5093	2702	115
Test 9	1799	5447	2925	44
Test 10	1853	6939	4677	98
(đơn vị thời gian là ms)				

- Đây là kết quả đo được thời gian chạy của 4 thuật toán sắp xếp trên theo đơn vị ms được thống kê.
- Dữ liệu đầu vào gồm 10 tệp tin chứa 1 triệu số nguyên, thực.
- Chạy trên thử nghiệm trên IDE PyCharm và các hàm sắp xếp được viết bằng ngôn ngữ Python.

## 2. Biểu đồ mô tả và phân tích đánh giá



- 1) Heap Sort (Chậm nhất): Tốn nhiều thời gian nhất, lên tới hơn 11.000ms ở một số bộ test. Lý do là vì cấu trúc Heap truy cập bộ nhớ không tuần tự cộng với chi phí gọi hàm đệ quy heapify quá nhiều trong môi trường thông dịch của Python.
- 2) Merge Sort (Ổn định): Thời gian thực thi dao động ổn định trong khoảng 2000ms - 6600ms. Mặc dù có độ phức tạp  $O(n \log n)$ , nhưng trong Python, việc phải cấp phát thêm bộ nhớ cho các mảng con khiến nó chạy chậm hơn Quick Sort đáng kể.
- 3) Quick Sort (Tốt nhất trong các thuật toán tự viết): Đạt thời gian trung bình khoảng 1800ms - 3800ms. Thuật toán này thể hiện hiệu suất tốt nhờ việc chọn Pivot ở giữa, giúp tránh được hiện tượng suy biến  $O(n^2)$  ở các bộ test có thứ tự đặc biệt.
- 4) NumPy Sort: Đây là thuật toán nhanh nhất trong mọi trường hợp, chỉ tốn khoảng 44ms - 115ms. Điều này là do thư viện NumPy được viết bằng ngôn ngữ C đã tối ưu hóa bộ nhớ đệm và sử dụng các thuật toán cực kỳ mạnh mẽ.