Lớp: KHTN2021

BÁO CÁO KẾT QUẢ THỬ NGHIỆM

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

Sinh viên thực hiện: Nguyễn Đức Nhân

Nội dung báo cáo: Nghiệm thu thời gian thực thi các thuật toán sort trên ngôn ngữ C++

I. Cấu hình phần cứng thực hiện thực nghiệm & bộ dữ liệu thực nghiệm:

1. Cấu hình phần cứng thực hiện thực nghiệm:

- CPU: Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz
- GPU: NVIDIA GeForce GTX 1660
- SSD: WD Blue SN550 1 TB M.2 NVMe PCIe Gen3 x4 (Read 2400 Mb/s, Write 1950 Mb/s)
- RAM: 15.92 GB RAM – 2933 MHz
- MAIN BOARD: B460M DS3H GIGABYTE

2. Bộ dữ liệu thực nghiệm:

- Các test case được khởi tạo ngẫu nhiên 10^6 giá trị theo phân phối chuẩn được cung cấp bởi thư viện Numpy trong Python. Các giá trị nằm trong khoảng -10^6 đến 10^6.
- Đặc tính các test case:
  + Từ test case 1 đến test case 8: Các giá trị được giữ nguyên thứ tự như mảng đã được sinh.
  + Test case 9: Các giá trị trong mảng được sắp xếp theo thứ tự tăng dần.
  + Test case 10: Các giá trị trong mảng được sắp xếp theo thứ tự giảm dần.

3. Lưu ý: Thực nghiệm chỉ khảo sát thời gian thực hiện thuật toán, không bao gồm thời gian đọc và trích xuất dữ liệu.

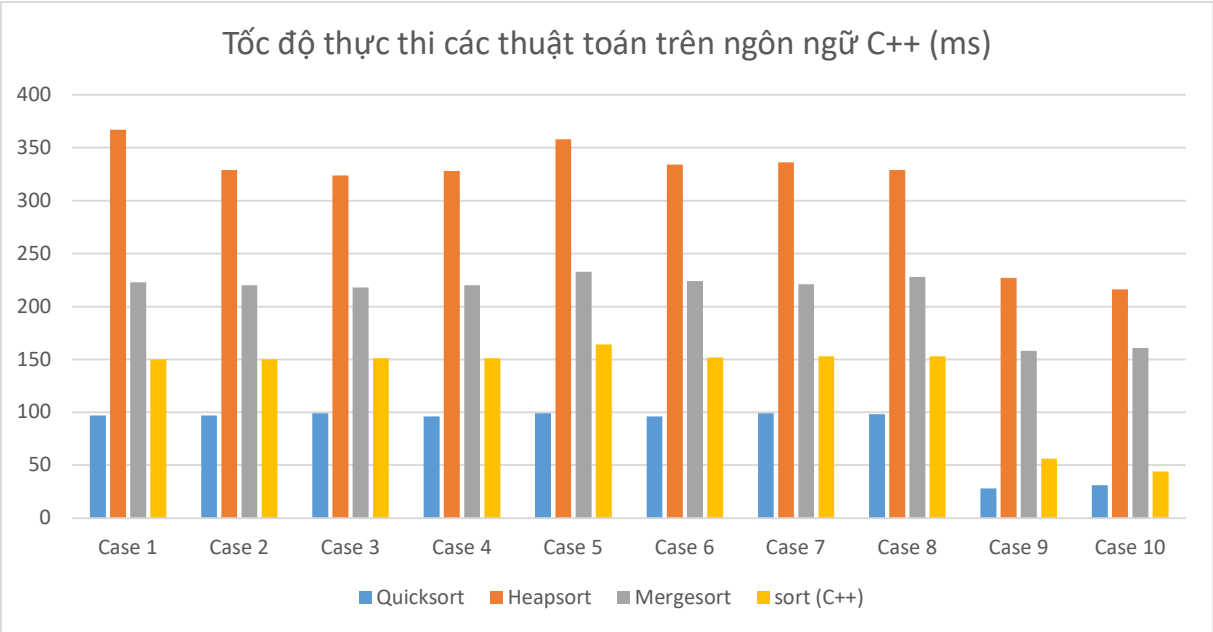
II. Kết quả thử nghiệm

1. Bảng thời gian thực hiện[1]

Dữ liệu	Thời gian thực hiện (ms)			
	Quicksort	**Heapsort**	**Mergesort**	**sort (C++)**
1	97	367	223	150
2	97	329	220	150
3	99	324	218	151
4	96	328	220	151
5	99	358	233	164
6	96	334	224	152
7	99	336	221	153
8	98	329	228	153
9	28	227	158	56
10	31	216	161	44
Trung bình	84	314.8	210.6	132.4

[1] Số liệu chỉ mang tính chất tham khảo do việc thực nghiệm bị ảnh hưởng bởi nhiều yếu tố.

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



III. *Kết luận:*

Đánh giá sơ bộ thì cả 4 thuật toán được sử dụng đều có độ phức tạp O(nlogn). Với thời gian thực thi trung bình của các thuật toán lần lượt là Quick sort – 84 ms, Heap sort – 314.8 ms, Merge sort – 210.6 ms và sort(C++) – 132.4 ms. Đánh giá tốc độ thực thi theo kết quả thực nghiệm: Quick sort < sort(C++) < Merge sort < Heap sort.

Các kết quả thực nghiệm cho thấy tốc độ thực thi thuật toán Quick sort với cải tiến Middle Pivot đem lại hiệu quả cao nhất ở cả bộ dữ liệu đặc biệt và các bộ dữ liệu phân phối chuẩn. Hai thuật toán Merge sort và sort(C++) đem lại kết quả có sự ổn định nhưng chậm hơn thuật toán Quick sort trên bộ dữ liệu biểu diễn dạng mảng. Và Heap sort có tốc độ thực thi chậm nhất.

Tuy nhiên phương pháp cài đặt cũng như số lượng bộ dữ liệu còn hạn chế nên kết quả cần đánh giá và cải thiện thêm trong thời gian tới.

IV. *Thông tin chi tiết – link github, trong repo gibub cần có*

1. Báo cáo: Các thông tin chi tiết bao gồm mã nguồn, báo cáo, các bộ dữ liệu xem thêm tại link github đính kèm bên dưới.
2. Link github: https://github.com/UIT-21520373/Data-structures-Algorithms