

BÁO CÁO THIẾT KẾ THUẬT TOÁN SONG SONG CHO SẮP XẾP DÃY TĂNG DẦN BẰNG THUẬT TOÁN MERGE SORT

Phan Nguyễn Hữu Phong - 22521090

Nguyễn Tiến Huy - 22520567

I. Giới thiệu

- Báo cáo tập trung vào việc phân tích và thiết kế thuật toán song song cho sắp xếp tăng dần bằng thuật toán **Merge Sort**.
- **Merge Sort** là một trong những thuật toán sắp xếp ổn định và hiệu quả nhất.
- Việc áp dụng thuật toán song song có thể giúp nâng cao hiệu suất xử lý, tối ưu tài nguyên và giảm thời gian xử lý.
- Chương trình minh họa và kết quả thực nghiệm ở:
[PhongPhan2k4/Parallel Merge Sort \(github.com\)](https://github.com/PhongPhan2k4/Parallel_Merge_Sort)

II. Lựa chọn cấu trúc dữ liệu

- Chọn cấu trúc dữ liệu để chia dãy số thành các phần nhỏ hơn để sắp xếp riêng lẻ và sau đó kết hợp chúng lại để được một dãy đã sắp xếp.
- Cấu trúc dữ liệu phù hợp là list, giúp dễ dàng lưu trữ và truy cập các phần tử.

III. Thiết kế thuật toán

- Thuật toán **Merge Sort** là một thuật toán sắp xếp dựa trên nguyên tắc “chia để trị”. Ý tưởng của thuật toán là chia dãy cần sắp xếp thành các phần nhỏ hơn để sắp xếp riêng lẻ, sau đó trộn chúng lại để được một dãy đã sắp xếp.
- Để thực hiện thuật toán song song, ta chia công việc sắp xếp các phần của dãy trong các tiến trình hoặc luồng riêng biệt. Sau đó, chúng ta trộn các phần đã sắp xếp để thu được dãy sắp xếp hoàn chỉnh.
- Tóm tắt các bước thực hiện:
 - Phân chia dãy ban đầu thành các mảng con phụ thuộc vào số lượng lõi hoặc tài nguyên có sẵn.
 - Sắp xếp từng mảng con bằng thuật toán **Merge Sort**. Các tiến trình hoặc luồng riêng biệt được sử dụng để sắp xếp các mảng con.
 - Trộn các mảng con để thu được mảng đã sắp xếp.
 - Sau khi quá trình trộn hoàn thành, ta thu được một dãy đã sắp xếp hoàn chỉnh.

- Trong Python, để thiết kế thuật toán song song cho Merge Sort. Ta có thể sử dụng thư viện:
 - **multiprocessing** để tạo các tiến trình song song.
 - **threading** để tạo các luồng song song.
- Ta có thể điều chỉnh số lượng tiến trình hoặc luồng theo tài nguyên của máy tính để tối đa hóa hiệu năng.

IV. Nhận xét và kết luận

- Thiết kế thuật toán song song cho phép các phần của thuật toán **của Merge Sort** được thực thi song song trên nhiều luồng, giúp giảm thời gian thực thi.
- Sử dụng nhiều luồng cho phép tận dụng tài nguyên tính toán đa lõi trên các nền tảng phần cứng.
- Thiết kế song song cho phép mở rộng khả năng xử lý của thuật **toán Merge Sort**, giúp xử lý các dãy số lớn và phức tạp một cách hiệu quả.
- Hiệu quả và tốc độ thực thi của thuật toán phụ thuộc vào tài nguyên của hệ thống đang sử dụng. Nên trong một vài trường hợp thuật toán song song có thể cho thời gian chạy chậm hơn.

V. Tài liệu tham khảo

- [Parallel Merge Sort \(opengenius.org\)](https://opengenius.org/parallel-merge-sort/)
- stackoverflow.com/questions/8435640/parallel-merge-sort-performance
- [Parallel Algorithm - Sorting \(tutorialspoint.com\)](https://tutorialspoint.com/parallel-algorithm-sorting/)
- [Parallel Merge Sort — Parallel Sorting \(macalester.edu\)](https://macalester.edu/parallel-merge-sort/)