ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN THỰC HÀNH
BÀI TOÁN SẮP XẾP DỮ LIỆU LỚN

Học phần: Cấu trúc dữ liệu và giải thuật

Lớp: 21CTT4

Họ và tên các thành viên:

1. **Ngô Nhật Tân – 19120128**
2. **Phạm Khánh Hoàng Việt – 20120626**

***Thành phố Hồ Chí Minh*, tháng 12 năm 2022**

Nội dung

I. Giới thiệu bài toán

- Trong đồ án này ta sẽ sắp xếp file dữ liệu không fit trên RAM. Nhiệm vụ là sắp xếp theo "The Id of Book" từ nhỏ đến lớn và lưu vào file "sorted_books_rating.csv".
- Bộ dữ liệu được lấy từ đường dẫn sau: https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews

II. Thành viên và mức độ hoàn thành công việc

- Nhóm có hai thành viên là:
  - Ngô Nhật Tân – 19120128
  - Phạm Khánh Hoàng Việt – 20120626

Công việc	Người thực hiện	Đóng góp	Mức độ hoàn thành công việc
Viết hàm ***SplitFile*** để chia từ file lớn ra thành các file nhỏ	Ngô Nhật Tân	100%	100%
Viết hàm ***MergeSort***	Ngô Nhật Tân	100%	100%
Viết hàm ***SortFile***	Ngô Nhật Tân	70%	100%
	Phạm Khánh Hoàng Việt	30%	
Viết các hàm kiểm tra và tạo File, Folder,… và các hàm nhỏ lẻ	Ngô Nhật Tân	100%	100%
Viết hàm ***MergeSortedFile*** để merge các file nhỏ thành file lớn	Phạm Khánh Hoàng Việt	100%	100%
Chia code thành những file riêng	Ngô Nhật Tân	100%	100%
Viết báo cáo	Ngô Nhật Tân	100%	100%

III. Kiến trúc và thuật toán sử dụng

1. Cấu trúc các file:

- **file_controller.h và file_controller.cpp**: chứa các hàm liên quan đến file như là **IsDirExist, CreateFile, SplitFile, DeleteFile,…**

- **sort.h và sort.cpp:** chứa các hàm liên quan đến sort như là **MergeSort, MergeSortedFile, SortFile,…**
- **review.h và review.cpp:** chứa class **Review** và các hàm **GetData, GetId,…**

2. Mô tả thuật toán

- Để giải quyết bài toán này tụi em dùng thuật toán **Merge Sort** để sắp xếp dữ liệu.
- Trước tiên sẽ split file lớn ra thành các file nhỏ và lưu vào thư mục **output**, mỗi file nhỏ sẽ có **10.000** dòng review, có thể thay đổi số lượng trong mỗi file nhỏ tại biến **SIZE** ở file ***file_controller.h***.
- Sau khi đã chia thành các file nhỏ thì sẽ dùng thuật toán Merge Sort để sắp xếp cho từng file và lưu các file đã sắp xếp vào thư mục **sorted**, sau đó sẽ xóa folder **output.**
- Khi đã sort xong thì sẽ dùng vào lặp for để gọi hàm:

```
void MergeSortedFile(string path1, string path2)
```

- Hàm **MergeSortedFile** sẽ load dữ liệu của 2 file truyền vào và xóa 2 file đó. Sau đó sẽ so sánh dữ liệu của 2 file đó và lưu vào file ***sorted_books_rating.csv*** (vẫn nằm trong folder sorted).
- Sau khi đã sort xong ta sẽ chuyển file này sang cùng thư mục với file ***Books_rating.csv*** và gọi hàm **AddHeader** để thêm header vào file, sau đó xóa thư mục **sorted.**

3. Độ phức tạp thuật toán

- Hàm **SplitFile** có độ phức tạp thuật toán là **O(n).**
- Hàm **MergeSort** có độ phức tạp thuật toán là **O(nlog(n)).**
- Hàm **SortFile** có độ phức tạp thuật toán là **O(n^2).**
- Hàm **MergeSortedFile** có độ phức tạp thuật toán là **O(n).**

  ⇨ Độ phức tạp thuật toán của bài này là **O(n^2).**

1. Kết luận

- Nhiệm vụ chính là sort file không fit trên RAM đã hoàn thành tuy nhiên việc thực hiện tốn khá nhiều thời gian và bộ nhớ.

2. Link source code

- Github: [tanngo2510/sort-big-data: Môn học: Cấu trúc dữ liệu và giải thuật (github.com)]