ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

* * * * *

BÀI TẬP
SẮP XẾP DỮ LIỆU LỚN

Nhóm thực hiện: 5 – **ANTIFRAGILE**
20120074 – Nguyễn Gia Hào
21120417 – Nguyễn Thị Ngọc Châm

Thành phố Hồ Chí Minh – 2022

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

* * * * *

BÀI TẬP
SẮP XẾP DỮ LIỆU LỚN

Môn học: *Thực hành Cấu trúc dữ liệu và Giải thuật*
Lớp: *21CTT4A*
Giảng viên: *Nguyễn Thái Vũ*
Nhóm thực hiện: 5 – **ANTIFRAGILE**
20120074 – Nguyễn Gia Hào
21120417 – Nguyễn Thị Ngọc Châm

Thành phố Hồ Chí Minh – 2022

1. Giới thiệu

- Nhóm 6 – ANTIFRAGILE
- Thành viên:
  - ❖ *20120074 – Nguyễn Gia Hào*
  - ❖ *21120417 – Nguyễn Thị Ngọc Châm*
- Bảng phân công:

	Công việc	**Mức độ hoàn thành**
20120074	Code, Viết báo cáo	100%
21120417	Chia nhỏ file 3GB, Code	100%

2. Phát biểu bài toán

- Bộ dữ liệu: https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews. Download file ***Books_rating.csv***
- Nhiệm vụ là sắp xếp theo *The Id of Book* từ nhỏ đến lớn. Trường hợp hai rating có *book id* bằng nhau, thì cứ sắp xếp ngẫu nhiên.
- Kết quả sắp xếp được lưu vào file ***sorted_books_rating.csv***

3. Mô tả kiến trúc và thuật toán sử dụng vào bài toán

- **Struct BooksRating** gồm 2 thuộc tính:
  - ➢ **string id:** dùng để sắp xếp.
  - ➢ **string book_other_information:** là chuỗi lưu những thông tin còn lại của sách, không phải tách riêng các thông tin này ra để đơn giản hóa việc đọc file mà vẫn xuất ra kết quả đúng.
- Nhóm sử dụng *vector* để lưu dữ liệu:
  - ➢ Sử dụng *vector hai chiều* ***vector<vector<BooksRating>>*** để lưu danh sách các *book* đọc được từ các file csv. Mỗi *vector* trong *vector hai chiều* sẽ lưu trữ danh sách các *book* đọc được từ 1 file csv.
  - ➢ Sử dụng 1 vector *result* để lưu danh sách các *book* sau khi đã đọc, sort mỗi file và ghép các file csv lại với nhau. Từ đó dử dụng *vector result* để ghi vào file kết quả.
  - ➢ Lý do chọn *vector:* là mảng động, thích hợp cho việc thêm phần tử mới vào mảng, nhất là với việc đọc nhiều file với số lượng phần tử mỗi file là khác nhau; tiện hơn con trỏ vì không cần quan tâm vấn đề rò rỉ bộ nhớ.
- Thuật toán sử dụng:
  - ➢ Thuật toán *Merge Sort* (hàm ***merge_sort***): có độ phức tạp trung bình là *O(nlogn)* giải quyết khá nhanh; có tính ổn định; xử lý khá tốt với dữ liệu lớn, đặc biệt là dạng *list, file* → phù hợp với dữ liệu của bài toán hiện tại.
  - ➢ Thuật toán *2 con trỏ* (hàm ***two_pointer***): thuật toán gộp 2 mảng đã sắp xếp để được mảng mới cũng đã được sắp xếp, dựa theo ý tưởng của thuật toán *Merge Sort*. Độ phức tạp là *O(m + n)* với *m* và *n* lần lượt là kích thước của 2 mảng.

4. Giải thích hàm

- Hàm ***read_multi_file():*** vì file nhóm chia ra chỉ có file đầu là có hàng tiêu đề, do đó code đọc file của hàm này sẽ chia ra 2 phần chính. Phần đọc file đầu tiên sẽ thêm 1 dòng gán dòng tiêu đề của file đầu vào biến *title_row*, phần còn lại sẽ đọc giống nhau giữa các file còn lại. Folder chứa các file csv này có tên là ***Books_rating*** sẽ nằm trong folder chứa project, trong folder này sẽ chứa 10 file csv nhỏ được tách từ file csv gốc, và đường dẫn đến các file này sẽ được lưu trong file ***local_name_file.txt*** để cùng folder với file ***Source.cpp*** chứa code, khi muốn lấy đường dẫn để đọc các file csv chỉ việc gọi đọc đến file ***local_name_file.txt*** để lấy đường dẫn.
- Hàm ***write_local_name_file():*** hàm này dùng để ghi đường dẫn đến các file csv vào file ***local_name_file.txt*** đã nêu ở trên. Thuộc tính **NUMBER_OF_FILE** ở dòng 9 định nghĩa cho số file đã chia từ file gốc (cụ thể của nhóm là 10 file). Nếu muốn chạy thử với dữ liệu ít hơn thì có thể thay đổi con số này ít lại.
- Hàm ***read_one_book():*** đọc thông tin 1 book.
- Hàm ***read_list_book():*** đọc 1 danh sách book của 1 file csv.
- Hàm ***merge() và merge_sort():*** thuật toán sắp xếp *Merge Sort.*
- Hàm ***sort_multi_list():*** sắp xếp các danh sách book trong mỗi *vector* theo *id*.
- Hàm ***two_pointer():*** kỹ thuật 2 con trỏ.
- Hàm ***merge_list():*** gộp các *vector* đã sắp xếp trong thành *vector* mới cũng đã được sắp xếp
- Hàm ***write_list():*** ghi kết quả ra file ***sorted_books_rating.csv***, file này sẽ nằm trong cùng thư mục với file ***local_name_file.txt*** và ***Source.cpp***.

NOTE: Folder ***Books_rating*** lúc nộp nhóm em sẽ để trống, nếu thầy muốn chạy code thì phải download 10 file csv về rồi chép vào thư mục này. Ngoài ra nhóm còn có bộ dữ liệu chạy thử với dung lượng nhỏ. Link download nhóm em sẽ để ở phần **Source code**.

5. Kết quả và kết luận

- Nhóm đã chạy thành công trên 3,000,000 dòng dữ liệu của file gốc.
- Thời gian chạy: ~40 phút.
- Kết luận: rèn luyện được khả năng sắp xếp file dữ liệu lớn, không fit trên RAM; rèn luyện được kỹ năng làm việc nhóm, kỹ năng lập trình.

6. Source code

- Source code: phần code nằm trong file ***Source.cpp*** được đặt trong folder ***Folder source code/20120074_21120417***. Ngoài ra nhóm còn đăng tải cả bài tập này lên trang Github: GitHub
- Link folder 10 file csv: Books_rating
- Link folder chứa file chạy thử: Books_rating_test