TRƯỜNG ĐẠI HỌC



**SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

HCMC University of Technology and Education

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**



NGÀNH CÔNG NGHỆ THÔNG TIN

**BÁO CÁO ĐỒ ÁN 1**

**THUẬT TOÁN MERGE SORT, EXTERNAL SORTING VÀ**

**ỨNG DỤNG**



|  |  |
| --- | --- |
| **Nhóm sinh viên thực hiện:** |  |
| **Lâm Hoàng An** | **18110073** |
| **Nguyễn Thị Như Quỳnh** | **18110189** |
|  |  |
| **GVHD: Th.S Lê Văn Vinh** |

**TP. Hồ Chí Minh, tháng 12 – 2020**

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Giáo viên hướng dẫn

(ký và ghi họ tên)

Ths. Lê Văn Vinh

LỜI CẢM ƠN

*Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Nay chúng em xin được phép gửi lời cảm ơn chân thành này đến thầy Lê Văn Vinh, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình định hướng chọn đề tài, hướng dẫn, nhận xét và góp ý cũng như cung cấp tài liệu tham khảo. Nếu không có những lời hướng dẫn, những kinh nghiệm thực tiễn của thầy thì chúng em nghĩ rằng bài thu hoạch này sẽ khó có thể hoàn thiện và hoàn thành đúng thời hạn được. Một lần nữa, chúng em xin cảm ơn thầy.*

*Chúng em cũng xin chân thành cảm ơn các quý thầy cô trong khoa Công Nghệ Thông Tin đã giúp đỡ hỗ trợ kiến thức cũng như giải đáp thắc mắc của chúng em. Cùng với đó,chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp cho chúng em hoàn thiện đề tài hơn.*

*Bài thu hoạch được thực hiện trong khoảng thời gian gần 7 tuần. Khoảng thời gian có hạn, cùng với kiến thức còn hạn chế và còn nhiều bỡ ngỡ khác do đó thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý Thầy Cô để kiến thức của chúng em được hoàn thiện hơn sau này. Chúng em xin chân thành cảm ơn.*

*Thành phố Hồ Chí Minh, ngày 14 tháng 12 năm 2020*

MỤC LỤC

[DANH MỤC CÁC HÌNH 1](#_Toc61789215)

[DANH MỤC CÁC BẢNG 1](#_Toc61789216)

[1. MÔ TẢ PROJECT 2](#_Toc61789217)

[1.1. Cơ sở lý thuyết: 2](#_Toc61789218)

[1.1.1. Thuật toán Merge Sort: 2](#_Toc61789219)

[2. QUÁ TRÌNH THỰC HIỆN PROJECT 3](#_Toc61789220)

[2.1. Thuật toán: 3](#_Toc61789221)

[2.1.1. Merge Sort: 3](#_Toc61789222)

[2.1.2. External Sort: 4](#_Toc61789223)

[2.2. Demo thuật toán: 4](#_Toc61789224)

[2.3. Kết luận: 5](#_Toc61789225)

[2.3.1. Nội dung: 5](#_Toc61789226)

[2.3.2. Project: 5](#_Toc61789227)

[3. PHÂN CÔNG CÔNG VIỆC 6](#_Toc61789228)

[4. TÀI LIỆU THAM KHẢO 6](#_Toc61789229)

# 

# DANH MỤC CÁC HÌNH

[Hình 1: Ví dụ về thuật toán Merge Sort 3](#_Toc61789280)

[Hình 2: Thuật toán Merge Sort 3](#_Toc61789281)

[Hình 3. Thuật toán External Sort 4](#_Toc61789282)

[Hình 4. Input 4](#_Toc61789283)

[Hình 5. Ouptut 5](#_Toc61789284)

# DANH MỤC CÁC BẢNG

[Bảng 1. Phân công công việc 6](#_Toc61789285)

# MÔ TẢ PROJECT

## Cơ sở lý thuyết:

### Thuật toán Merge Sort:

Merge Sort (sắp xếp trộn) là một thuật toán để sắp xếp các danh sách theo một trật tự nào đó. Thuật toán này là một ví dụ điển hình cho thuật toán Chia để trị. Merge Sort chia mảng đầu vào thành 2 nửa. Gọi đệ quy cho 2 danh sách vừa được chia cho đến khi không thể chia được nữa. Sau đó trộn 2 danh sách vừa được chia theo một quy luật nhất định.

* + 1. Thuật toán External Sort:

External Sort là một thuật toán sắp xếp có thể xử lý một lượng lớn dữ liệu. External Sort được yêu cầu khi dữ liệu sắp xếp không vừa với bộ nhớ chính của máy tính (RAM) và thay vào đó, chúng phải nằm trong bộ nhớ ngoài chậm hơn (ổ cứng). External thường sử dụng Merge Sort. Trong quá trình sắp xếp, các phần dữ liệu đủ nhỏ để vừa trong bộ nhớ chính sẽ được đọc, sắp xếp và ghi ra một tệp tạm thời. Sau khi Merge, các tệp con vừa được sắp xếp sẽ được kết hợp thành một tệp lớn hơn.

Điều kiện tiên quyết cho thuật toán:

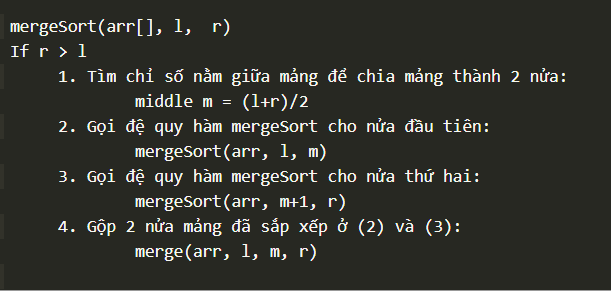
* Merge Sort: được sử dụng để sắp xếp các lần chạy riêng lẻ (một lần chạy là một phần của tệp đủ nhỏ vừa bộ nhớ chính)
* Merge K Sorted Array: được sử dụng để hợp nhất các tệp riêng lẻ đã được sắp xếp thành 1 tệp lớn hơn.

# QUÁ TRÌNH THỰC HIỆN PROJECT

## **Thuật toán**:

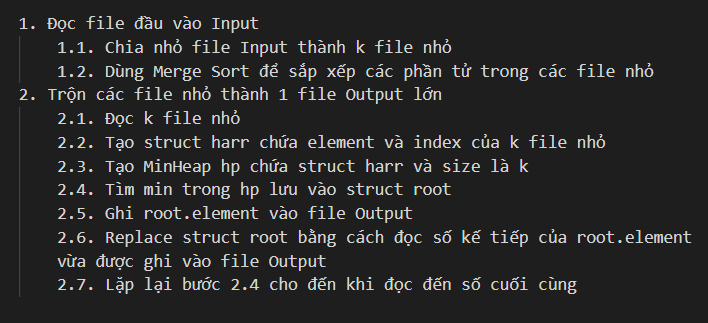
### Merge Sort:

#### Hình 1: Ví dụ về thuật toán Merge Sort



#### Hình 2: Thuật toán Merge Sort

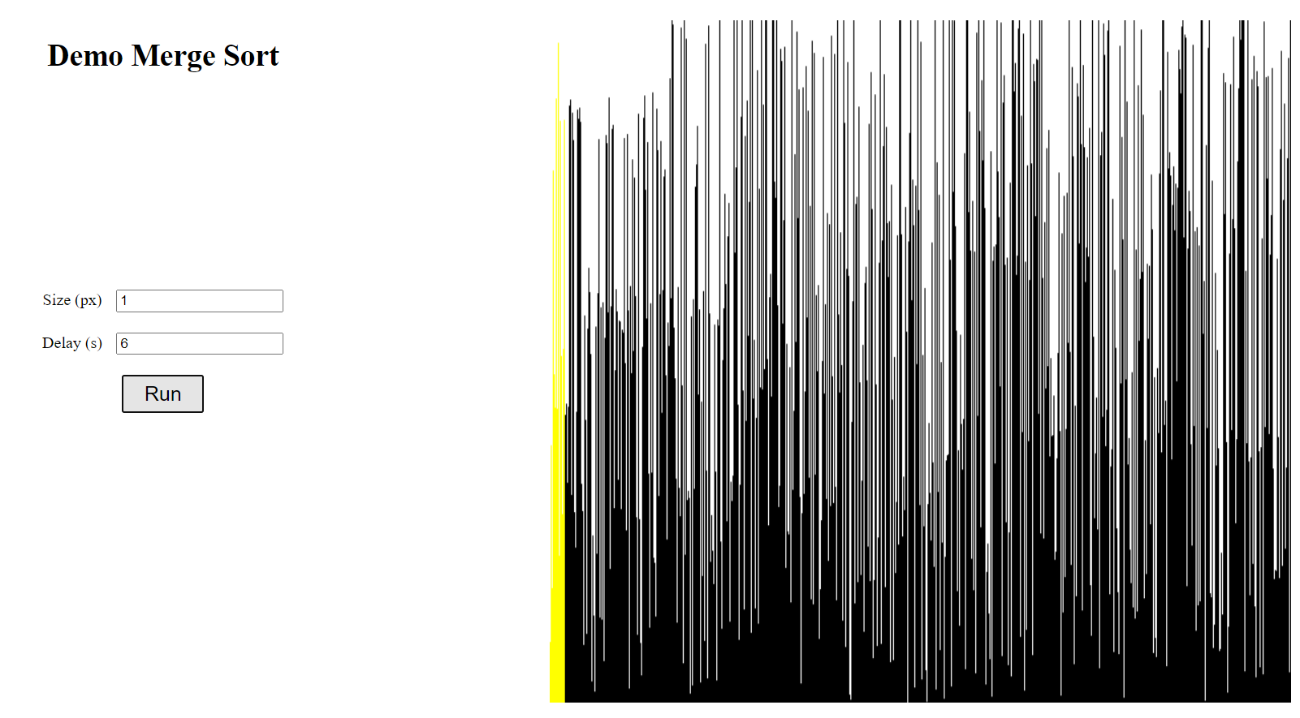
### External Sort:



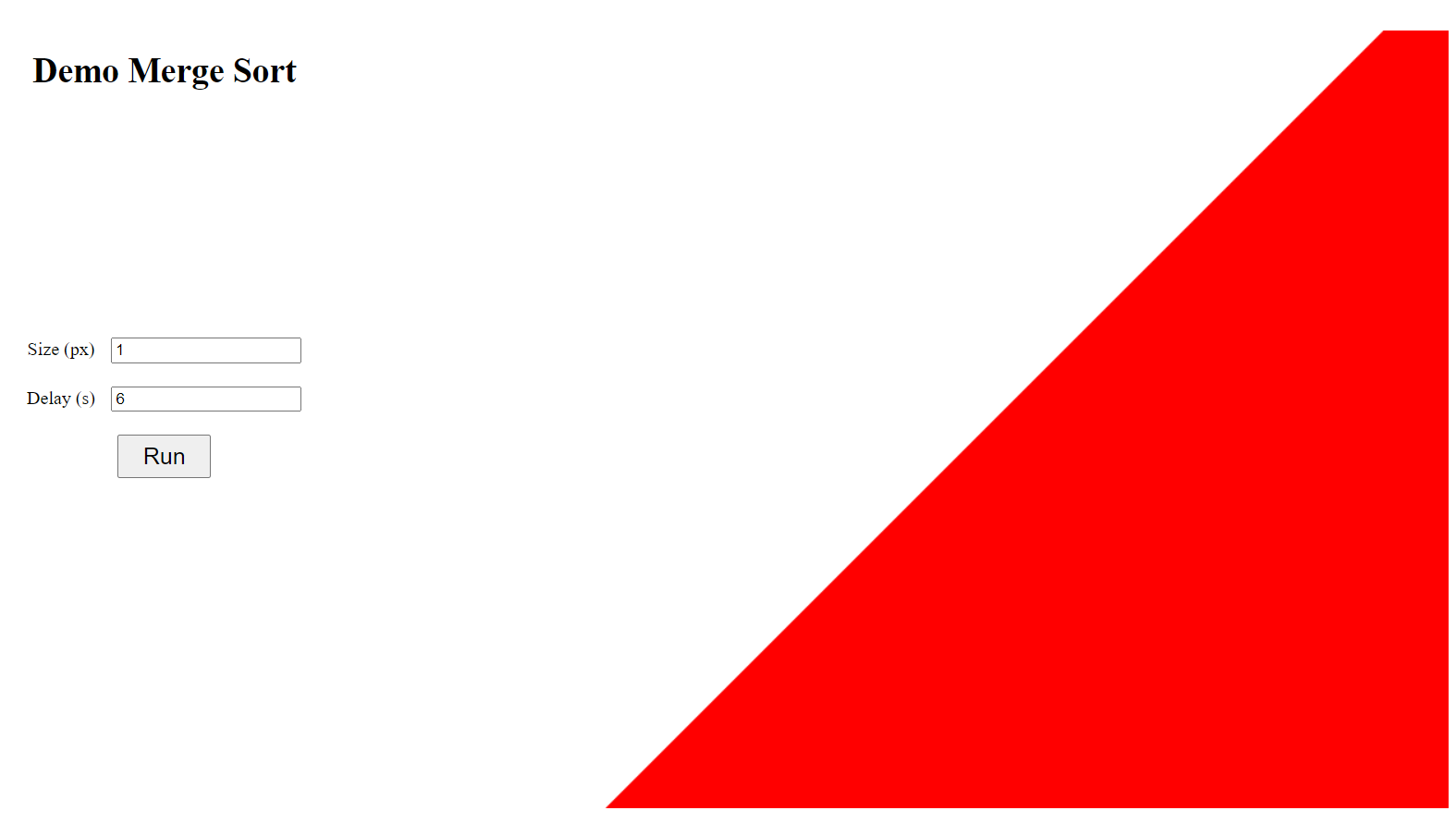
#### Hình 3. Thuật toán External Sort

## Demo thuật toán:

Sử dụng HTML/CSS để demo 2 thuật toán trên.



#### Hình 4. Input



#### Hình 5. Ouptut

## Kết luận:

### Nội dung:

Thuật toán Merge Sort là một trong những thuật toán sắp xếp nhanh được sử dụng rộng rãi, độ phức tạp thời gian là O = nlogn.. Merge Sort có tốc độ ổn định .Vì vậy sử dụng Merge Sort mang lại hiệu quả cực kì lớn.

Thuật toán External Sort có độ phức tạp thời gian là O = n + run\_size\*logrun\_size.. Nó thường được sử dụng cho các bài toán sắp xếp lớn, bộ nhớ chính của máy không thể chứa đủ 1 file đầu vào lớn.

### Project:

* Ưu điểm:
* Hoàn thành 100% theo yêu cầu của đề tài
* Hiểu được 2 thuật toán Merge Sort và External Sort
* Khuyết điểm: chưa tìm hiểu sâu rộng hơn về thuật toán External Sort

# PHÂN CÔNG CÔNG VIỆC

##### Bảng 1. Phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên SV | Mô tả công việc | Đóng góp |
| 1 | Lâm Hoàng An | - Viết thuật toán External Sort  - Viết demo giao diện cho thuật toán | 50% |
| 2 | Nguyễn Thị Như Quỳnh | - Viết thuật toán Merge Sort  - Kiểm thử chương trình  - Viết báo cáo | 50% |

# TÀI LIỆU THAM KHẢO

[1] Introduction to Design and Analyst of Algothrism – *Anany Levitin*

[2] https://www.geeksforgeeks.org/merge-sort/

[3] https://www.geeksforgeeks.org/external-sorting/