**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

🞠◊🞠◊🞠

****

**Tiểu luận cuối kỳ**

**Đề tài: Thuật Toán Merge Sort**

**Môn học: Đồ án CNTT**

**MÃ SỐ LỚP HP:**

**GVHD: TS. Lê Văn Vinh**

**NHÓM THỰC HIỆN: NHÓM 06**

**HỌC KỲ: 1 –NĂM HỌC: 2022 – 2023**

**Sinh viên thực hiện: Đỗ Ngọc Duy Hưng MSSV:20145267**

**Nguyễn Vũ Hồng Khánh MSSV:19151241**

**TP. HỒ CHÍ MINH – THÁNG 11 /NĂM 2022**

**Họ tên sinh viên thực hiện đề tài:**

**1. Đỗ Ngọc Duy Hưng - MSSV: 20145267**

**2. Nguyễn Hồng Vũ Khánh - MSSV: 19151241**

|  |  |
| --- | --- |
| ***Họ và tên*** | ***Công Việc*** |
| Đỗ Ngọc Duy Hưng | Viết báo cáo, code giao diện,chạy dữ liệu,tìm hiểu thuật toán |
| Nguyễn Hồng Vũ Khánh | Viết báo cáo, code giao diện,chạy dữ liệu,tìm hiểu thuật toán |

**ĐIỂM SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

**NHẬN XÉT CỦA GV:**

Tp. Hồ Chí Minh, ngày …., tháng 11, năm 2022

Giảng viên ký tên

# LỜI CẢM ƠN

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, Thầy Lê Văn Vinh, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm ơn thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý giúp chúng em hoàn thành đề tài đúng hạn.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án. Do đó, trong quá trình làm nên đề tài có những 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 và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

Mục Lục

Phần 1 Giới Thiệu chung…………………………………………………………….

I Lý do chọn đề tài…………………………………………………………………...

II Mục đích của thuật toán…………………………………………………………

Phần 2 Nội dung thuật toán…………………………………………………………

I Giới thiệu về thuật toán……………………………………………………………

II Ý tưởng triển khai thuật toán……………………………………………………...

III Xây dựng thuật toán………………………………………………………………

IV Đánh giá thuật toán……………………………………………………………….

V Mô tả thuật toán bằng giao diện…………………………………………………..

Phần 3 Chạy dữ liệu thực……………………………………………………………

Phần 4 Kết Luận……………………………………………………………………

**Phần 1 Giới thiệu chung**

Ngày nay, CNTT đã và đang đóng vai trò quan trọng trong đời sống kinh tế, xã

hội của nhiều quốc gia trên thế giới, là một phần không thể thiếu trong xã hội hiện đại 4.0 năng động, ngày càng hiện đại hoá. Vì vậy, việc tin học hoá vào một số lĩnh vực là hoàn toàn có thể và phù hợp với xu hướng hiện nay.

**I Lý do chọn đề tài**

Thuật toán là 1 phần quan trọng không thể thiếu của CNTT, thuật toán giống như bản đồ dẫn đường giúp ta giải quyết được các bài toán dễ dàng. Trong thuật toán có nhiều loại về thuật toán như: Sắp xếp, tìm kiếm,…

Đề tài mà nhóm chúng em chọn hôm nay là thuật toán sắp xếp Merge Sort. Thuật toán sắp xếp là 1 phần quan trọng trong kỹ thuật lập trình để giúp tối ưu hóa bài toán cần giải quyết. Đó là lý do mà nhóm chúng em chọn đề tài thuật toán Merge Sort.

**II Mục đích của thuật toán**

Thuật toán Merge Sort:dùng để sắp xếp 1 mảng ban đầu bằng cách tách mảng ban đầu thành những mảng nhỏ hơn và sắp xếp chúng. Sau khi sắp xếp thì gộp những mảng đã sắp xếp lại thành 1 mảng mới đã được sắp xếp theo thứ tự.

**Phần 2 Nội dung thuật toán**

**I Giới thiệu về thuật toán**

Thuật toán Merge Sort: là thuật toán sắp xếp thuộc loại sắp xếp nhanh trong khoa học máy tính. Merge sort là thuật toán điển hình cho tư tưởng chia để trị để giải quyết các bài toán có dữ liệu lớn và phức tạp. Cụ thể với bài toán sắp xếp, nó sẽ chia nhỏ danh sách cần sắp xếp thành từng phần tử rời sau đó hòa nhập theo phương pháp trộn tự nhiên thành dãy có thứ tự

Độ khó của thuật toán Merge Sort là: O( n logn)

Độ phức tạp thời gian của Merge Sort trong trường hợp tốt nhất là: O

Độ phức tạp thời gian của Merge Sort trong trường hợp trung bình là: O( n logn)

Độ phức tạp thời gian của Merge Sort trong trường hợp sấu nhất là: O( n logn)

**II Ý tưởng triển khai thuật toán**

Các thuật toán sắp xếp đơn giản như Bubble Sort, Insertion Sort . . . đều không thể xử lý được dữ liệu lớn. Thuật toán sắp xếp trộn lấy ý tưởng từ việc chia để trị để chia nhỏ bài toán thành các bài toán nhỏ hơn, sau đó giải quyết chúng. Từ đó sẽ giúp xử lý dữ liệu lớn một cách tốt hơn, tối ưu về mặt thời gian.

Ý tưởng đưa ra như sau:

Chia danh sách gồm n phần tử chưa được sắp xếp thành n danh sách con, mỗi danh sách chứa một phần tử (danh sách một phần tử được coi là đã sắp xếp).Liên tục hợp nhất các danh sách con để tạo ra các danh sách con được sắp xếp mớ cho đến khi chỉ còn lại một danh sách. Đây sẽ là danh sách được sắp xếp.

Khi triển khai code, ta sẽ cụ thể hóa bằng các bước:

Bước 1:

Chia dãy cần sắp xếp thành 2 dãy con Từ dãy con thu được lại tiếp tục chia thành 2 dãy con nhỏ hơn nữaQuá trình phân chia tiếp tục cho đến khi thu được dãy con chỉ còn duy nhất 1 phần tử.

Bước 2:

Hòa nhập 2 dãy con nhỏ nhất thành dãy con lớn hơn sao cho đúng thứ tựTừ hai dãy con lớn hơn lại hòa nhập thành 2 dãy con lớn hơn nữa….Quá trình hòa nhập cứ tiếp tục như vậy cho đến khi thu được dãy số ban đầu đã được sắp xếp.

Để dễ hình dung ta xem hình ảnh dưới đây

Diagram

Description automatically generated

Lưu đồ thuật toán:

Diagram

Description automatically generated

**III Xây dựng thuật toán**

Dưới đây là code được viết bằng C++

Mersort sắp xếp theo chiều tăng dần

Text

Description automatically generated

Chia mảng ban đầu thành 2 mảng nhỏ hơn mảng 1 có n1 phần tử và mảng 2 có n2=n-n1 phần tử

Sau đó đưa các phần tử ban đầu vào 2 mảng con tiếp theo gọi đệ quy để chia mảng cho đến khi mảng có 1 phần tử rồi lần lượt so sánh 2 mảng để ráp lại thành mảng ban đầu.

Hàm main:

Text

Description automatically generated

**IV Đánh giá thuật toán**

Bảng đánh giả thuật toán:

Trường hợp Độ phức tạp Bộ nhớ sử dụng

Tốt nhất O (n log n) n

Trung bình O (n log n) n

Xấu nhất O(n log n) n

Ưu điểm:

Độ phức tạp trung bình O(n log n), tốc độ giải quyết khá nhanhCó tính ổn định và thích ứng, tốc độ không bị ảnh hưởng nhiều bởi dữ liệu đầu vàoXử lý khá tốt với dữ liệu lớn đặc biệt là dạng list, file

**Nhược điểm:**

Tốn nhiều bộ nhớ nếu sử dụng đệ quy Code khó cài đặt, tương đối phức tạp Trong hầu hết các trường hợp, thuật toán này không được đánh giá cao hơn Quick sort

Ứng dụng: Merge Sort rất hữu ích để sắp xếp các danh sách được liên kết trong thời gian O (nLogn). Trong trường hợp danh sách được liên kết, trường hợp này khác nhau chủ yếu do sự khác biệt trong phân bổ bộ nhớ của mảng và danh sách được liên kết. Không giống như mảng, các nút danh sách liên kết có thể không liền kề trong bộ nhớ. Không giống như một mảng, trong danh sách liên kết, chúng ta có thể chèn các mục vào giữa trong O (1) không gian thừa và O (1) thời gian. Do đó hoạt động hợp nhất của sắp xếp hợp nhất có thể được thực hiện mà không có thêm dung lượng cho danh sách được liên kết. Trong mảng, chúng ta có thể thực hiện truy cập ngẫu nhiên khi các phần tử nằm kề nhau trong bộ nhớ. Giả sử chúng ta có một mảng A số nguyên (4 byte) và đặt địa chỉ của A [0] là x thì để truy cập A [i], chúng ta có thể truy cập trực tiếp vào bộ nhớ tại (x + i \* 4). Không giống như mảng, chúng ta không thể thực hiện truy cập ngẫu nhiên trong danh sách liên kết. Sắp xếp nhanh yêu cầu rất nhiều loại truy cập. Trong danh sách liên kết để truy cập chỉ mục thứ i, chúng ta phải di chuyển từng nút từ đầu đến nút thứ i vì chúng ta không có khối bộ nhớ liên tục. Do đó, chi phí tăng đối với nhanh chóng. Sắp xếp hợp nhất truy cập dữ liệu một cách tuần tự và nhu cầu truy cập ngẫu nhiên thấp.

Vấn đề đảo ngược số lượng

Được sử dụng trong sắp xếp bên ngoài

**IV Mô tả thuật toán bằng giao diện**

Ở đây khi ta muốn nhập vào 1 mảng chưa sắp xếp và muốn sắp xếp lại nó thì chúng ta sẽ hiểu rõ hơn về cách thức hoạt động của nó ở đây chúng em mô tả thuật toán bằng trang web. Khi chạy thuật toán chúng ta sẽ thấy cách nó sắp xếp theo từng bước và ta sẽ hiểu rõ hơn về nó.

Ở đây chúng em minh họa bằng website

Trang web này có 2 phần

Phần 1

Graphical user interface, text, application

Description automatically generated

Giúp người dùng có thể nhập vào mảng cần sắp xếp và thời gian chạy cho mỗi bước

Phần 2 Minh họa cho mỗi bước chạy của chương trình ở đây có các button Sort là sắp xếp khi ta nhấn nút này thì mảng của chúng ta sẽ chạy từng bước và ta sẽ thấy được.

A screenshot of a computer screen

Description automatically generated with medium confidence

Một ví dụ mẫu: mảng gồm 8 phần tử [38,27,43,3,9,82,10] khi ta nhấn Sort thì mảng sẽ chạy từng bước cho ta thấy và kết quả sẽ là:

A screenshot of a computer

Description automatically generated with medium confidence

Để minh họa rõ hơn ta nên xem cách chạy của chương trình bên dưới

Graphical user interface, text, application

Description automatically generated

Cấu trúc của file dạo diện gồm 3 folder:

Folder js dùng để viết code thuật toán Merge Sort và các thao tác di chuyển khi sắp xếp.

Trogn file animation.js

Text

Description automatically generated

Đoạn code này khi chạy thuật toán nó sẽ dịch chuyển theo chiều y từ trên xuống 1 đoạn là 20px

Text

Description automatically generated

Text

Description automatically generated

Thuật toán Merge Sort khi nhấn nút Sort

Trong file array.js

Text

Description automatically generated

Đoạn code này giúp ta hiển thị các mảng vừa nhập vào dưới dạng giao diện để hiển thị

File main.js

Text

Description automatically generated

Sử lý sự kiện cho 2 nút là Sort và unsort

File setting.js

Folder sass dùng để css cho trang web trở nên đẹp hơn vì ở đây là project nhỏ nên nhóm em dùng css để thuận lợi và dễ dàng code hơn.

Folder Vendor dùng để lấy các icon cho trang web

Diagram

Description automatically generated

Ở đây mỗi dòng là mỗi bước chạy của thuật toán.

**Phần 3 Chạy dữ liệu thực**

Ở đây ta chạy bằng command bước đầu ta cần nhập số phần tử của mảng cần sắp xếp.

**Text

Description automatically generated with medium confidence**

Sau đó ta lần lượt nhập từng phần tử vào

**Text

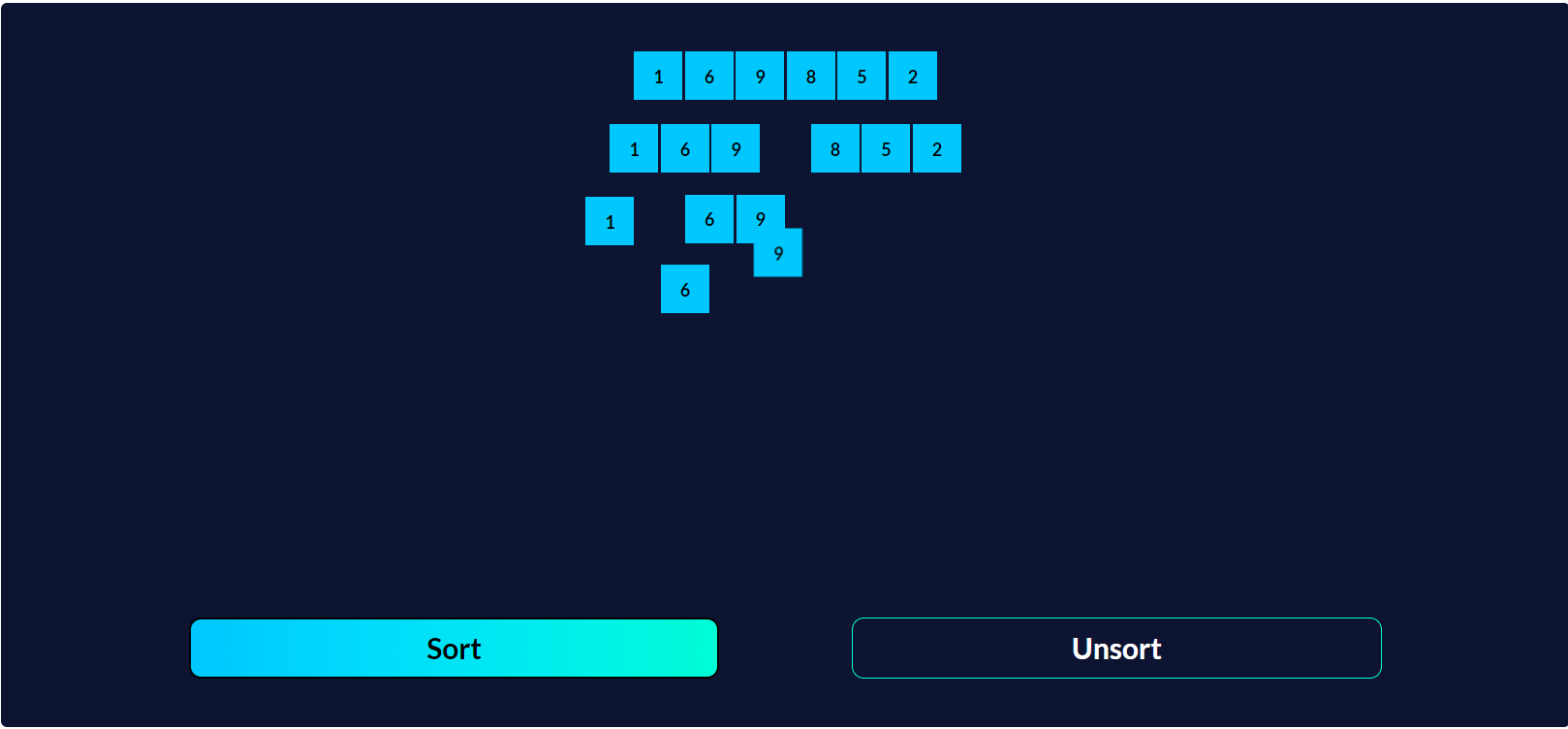
Description automatically generated**

Sau khi nhập đến phần tử cuối cùng của mảng thì nó sẽ xuất hiện mảng đã được sắp xếp theo thứ tự tăng dần

**Text

Description automatically generated**

Như vậy mảng ban đầu của chúng ta không có thứ tự đã thành mảng có thứ tự đã được sắp xếp



Quá trình thuật toán chạy bằng giao diện

**Phần 4 Kết luận**

Với mong muốn lập trình viên có thể hiểu rõ hơn về thuật toán Merge Sort và ứng dụng rộng rãi nó vào các dự án mà các lập trình viên tham gia để tối ưu hóa bài toán. Qua đề tài trên nhóm chúng em đã biết cách sử dụng và ứng dụng thuật toán Merge Sort vào các dự án mà chúng em làm.

**Tài liệu tham khảo**: W3School.com