BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**

**MÔ PHỎNG QUÁ TRÌNH SẮP XẾP MẢNG**

**BẰNG GIẢI THUẬT SHELL SORT**

**Giảng viên hướng dẫn: Đoàn Vũ Thịnh**

**Sinh viên thực hiện: Ngô Hoàng Vũ**

**Mã số sinh viên: 63131717**

KHÁNH HÒA - 2023

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ BÁO CÁO THỰC TẬP CƠ SỞ**

**Tên đề tài: MÔ PHỎNG QUÁ TRÌNH SẮP XẾP MẢNG BẰNG GIẢI THUẬT SHELL SORT**

**Giảng viên hướng dẫn:** Đoàn Vũ Thịnh

**Sinh viên được hướng dẫn:** Ngô Hoàng Vũ

**MSSV:** 63131717

**Khóa:** 63 **Ngành:** Công nghệ Thông tin

|  |  |  |  |
| --- | --- | --- | --- |
| **Lần** | **Ngày** | **Nội dung** | **Nhận xét của GVHD** |
| 1 | 11/12/2023 | Thông báo cho GVHD về đề tài đã chọn. Sinh viên trình bày sơ bộ chương trình demo. |  |
| 2 | 18/12/2023 | Sinh viên trình bày việc mô phỏng thuật toán chính dựa trên kiến thức đã được học ở môn kỹ thuật đồ họa và các kiến thức thu nhận được từ Internet để mô phỏng thuật toán. |  |
| 3 | 28/12/2023 | Sinh viên hoàn thiện mô phỏng thuật toán đã chọn |  |
| 4 | 05/01/2023 | Sinh viên nộp bản thảo của báo cáo thực tập. |  |

**Nhận xét chung (sau khi sinh viên hoàn thành ĐA/KL)**:

Điểm hình thức: /10 Điểm nội dung: /10 **Điểm tổng kết: /10**Đồng ý cho sinh viên: Được bảo vệ: 🗹 Không được bảo vệ: 

Khánh Hòa, ngày 29 tháng 12 năm 2023

Cán bộ hướng dẫn

*(Ký và ghi rõ họ tên)*

# LỜI CẢM ƠN

Sau thời gian nghiên cứu, học tập Trường Đại học Nha Trang và 8 tuần thực tập cơ sở, được sự giúp đỡ của quý thầy cô, bạn bè, các anh chị và cán bộ hướng dẫn thực tập tại công ty em đã hoàn thành báo cáo thực tập cơ sở học kỳ 3, năm học 2022-2023.

Để hoàn thành khóa thực tập này, em xin gởi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường Đại học Nha Trang, đặc biệt là thầy Đoàn Vũ Thịnh đã tận tình giảng dạy giúp cho em có được nhiều kiến thức và đã nhiệt tình hướng dẫn hướng dẫn em hoàn thành tốt khóa thực tập.

Trong quá trình thực tập, cũng như là trong quá trình làm bài báo cáo, khó tránh khỏi sai sót, rất mong các Thầy, Cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp Thầy, Cô để em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn trong các bài báo cáo của khóa học sắp tới. Em xin chân thành cảm ơn!

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc155525109)

[MỤC LỤC ii](#_Toc155525110)

[TÓM TẮT iii](#_Toc155525111)

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc155525112)

[1.1. Thuật toán sắp xếp 2](#_Toc155525113)

[1.2 Một số phương pháp sắp xếp thông dụng 2](#_Toc155525114)

[1.2.1 Bubble Sort 2](#_Toc155525115)

[1.2.2 Quick Sort 2](#_Toc155525116)

[1.3. Thuật toán sắp xếp Shell Sort 3](#_Toc155525117)

[1.3.1. Ý tưởng chính của thuật toán 3](#_Toc155525118)

[1.3.2 Lưu đồ thuật toán 4](#_Toc155525119)

[1.3.3 Đánh giá thuật toán 4](#_Toc155525120)

[1.3.3.1 Độ phức tạp 4](#_Toc155525121)

[1.3.3.2 Ưu điểm và nhược điểm 5](#_Toc155525122)

[1.3.3.3 Tốc độ thực hiện 6](#_Toc155525123)

[1.4. Dev C++ 6](#_Toc155525124)

[1.5. Thư viện Graphics.h 7](#_Toc155525125)

[CHƯƠNG 2. PHƯƠNG PHÁP NGHIÊN CỨU 7](#_Toc155525126)

[2.1. Cài đặt Dev C và thư viện graphics.h 7](#_Toc155525127)

[2.2. Cài đặt thuật toán Shell Sort 8](#_Toc155525128)

[CHƯƠNG 3. KẾT QUẢ THỰC HIỆN 13](#_Toc155525129)

[CHƯƠNG 4: THẢO LUẬN 14](#_Toc155525130)

[TÀI LIỆU THAM KHẢO 15](#_Toc155525131)

# **TÓM TẮT**

Trong khoa học máy tính, thuật toán sắp xếp là một thuật toán đặt các phần tử của danh sách thành một thứ tự. Những loại thứ tự được sử dụng thường xuyên nhất là thứ tự số và thứ tự từ điển, và tăng dần hoặc giảm dần. Sắp xếp hiệu quả rất quan trọng để tối ưu hóa hiệu quả của các thuật toán khác (như thuật toán tìm kiếm và hợp nhất) yêu cầu dữ liệu đầu vào phải nằm trong danh sách được sắp xếp. Sắp xếp cũng thường hữu ích cho việc kinh điển hóa dữ liệu và để tạo ra đầu ra có thể đọc được của con người. Quy trình thực hiện được trải qua các bước từ cài đặt thuật toán, hiển thị kết quả đầu ra trên màn hình đều được thực hiện trên môi trường C/C++ thông qua phần mềm DevC++ có kết hợp với thư viện graphics.h.

Sản phẩm đã minh họa được từng bước sắp xếp mảng sử dụng giải thuật sắp xếp Shell Sort. Đồng thời cũng chỉ ra các trường hợp hạn chế của mỗi thuật toán và cách khắc phục các nhược điểm đó

# CHƯƠNG 1. GIỚI THIỆU

Trong khoa học máy tính và trong toán học, thuật toán sắp xếp là một thuật toán mà sắp xếp các phần tử của một danh sách (hoặc một mảng) theo thứ tự (tăng hoặc giảm). Người ta thường xét trường hợp các phần tử cần sắp xếp là các số. Bài toán sắp xếp đã được nhiều nhà khoa học quan tâm. Từ thuật toán sắp xếp chúng ta có nhiều ứng dụng trong thực tế như: Danh sách lớp với các học sinh được sắp xếp theo thứ tự bảng chữ cái, danh bạ điện thoại, danh sách các truy vấn được tìm kiếm nhiều nhất trên Google…

Sắp xếp xuất hiện trong bất kỳ lĩnh vực nào của tin học, từ những ứng dụng ẩn bên trong của hệ điều hành như bài toán điều khiển quá trình, bài toán lập lịch cho CPU, bài toán quản lý bộ nhớ… cho đến những ứng dụng thông thường như sắp xếp dãy các từ, các câu, các bản ghi theo thứ tự nào đó.

## 1.1. Thuật toán sắp xếp

Sắp xếp là quá trình xử lý một danh sách các phần tử (hoặc các phần tử trong mảng) để đặt chúng theo một thứ tự, thỏa mãn một tiêu chuẩn nào đó dựa trên nội dung thông tin lưu giữ tại mỗi phần tử.

Cho trước một dãy số a1, a2,…, aN được lưu trữ trong cấu trúc dữ liệu mảng. Sắp xếp dãy số a1, a2,…, aN là thực hiện việc bố trí lại các phần tử sao cho được dãy mới ak1, ak2,…, akN được hình thành có thứ tự (giả sử xét thứ tự tăng) nghĩa là aki > aki – 1... Hai thao tác so sánh và gán là các thao tác cơ bản của hầu hết các thuật toán sắp xếp.

Khi xây dựng một thuật toán sắp xếp, cần tìm cách giảm thiểu những phép so sánh và đổi chỗ không cần thiết để tăng hiệu quả của thuật toán. Ðối với các dãy số được lưu trữ trong bộ nhớ chính, nhu cầu tiết kiệm bộ nhớ được đặt nặng, do vậy những thuật toán sắp xếp đòi hỏi cấp phát thêm vùng nhớ để lưu trữ dãy kết quả ngoài vùng nhớ lưu trữ dãy số ban đầu thường ít được quan tâm. Thay vào đó, các thuật toán sắp xếp trực tiếp trên dãy số ban đầu – gọi là các thuật toán sắp xếp tại chỗ – lại được đầu tư phát triển.

## 1.2 Một số phương pháp sắp xếp thông dụng

### 1.2.1 Bubble Sort

Đi từ cuối mảng về đầu mảng, trong quá trình đi, nếu phần tử ở dưới (đứng phía sau) nhỏ hơn phần tử đứng ngay trên (trước) nó thì theo nguyên tắc của bọt khí, phần tử nhẹ hơn sẽ bị “trồi” lên trên phần tử nặng (hai phần tử này sẽ được đổi chỗ cho nhau). Kết quả là phần tử nhỏ nhất (nhẹ nhất) sẽ được đưa về đầu dãy rất nhanh. Sau đó sẽ không xét đến nó ở bước tiếp theo, do vậy ở lần xử lý thứ i sẽ có vị trí đầu dãy là i. Lặp lại xử lý trên cho đến khi không còn cặp phần tử nào để xét và ta có dãy sắp xếp theo một thứ tự.

### 1.2.2 Quick Sort

Tính từ khóa chốt, các phần tử nhỏ hơn khóa chốt phải được xếp trước chốt; mọi phần tử lớn hơn khóa chốt được xếp vào sau chốt. Để làm được điều đó, các phần tử trong danh sách sẽ được so sánh với khóa chốt và tráo vị trí cho nhau hoặc cho khóa chốt nếu phần tử đó lớn hơn chốt mà lại nằm trước chốt hoặc nhỏ hơn chốt nhưng lại nằm sau chốt. Khi việc đổi chỗ lần đầu tiên đã được thực hiện xong thì danh sách tạo thành hai đoạn: một đoạn gồm các phần tử nhỏ hơn chốt, một đoạn gồm các phần tử lớn hơn chốt còn khóa chốt chính là vị trí của phần tử trong danh sách được sắp xếp. Áp dụng kỹ thuật như trên cho mỗi đoạn trước chốt và sau chốt cho tới khi các đoạn còn lại hai phần tử thì dừng lại. Khi đó, danh sách đã được sắp xếp.

Cụ thể giả sử để sắp xếp dãy số a gồm n phần tử: a1, a2,…, an; giải thuật Quick Sort dựa trên việc phân hoạch dãy ban đầu được chia thành hai phần:

* Dãy con 1 gồm các phần tử: a1,…, ai – 1 có giá trị nhỏ hoặc bằng ai;
* Dãy con 2 gồm các phần tử: ai + 1,…, an có giá trị lớn hơn ai;

Với ai là một phần tử bất kỳ trong dãy.

Để sắp xếp dãy con 1 và 2, lần lượt tiến hành việc phân hoạch từng dãy con theo cùng phương pháp phân hoạch dãy ban đầu.

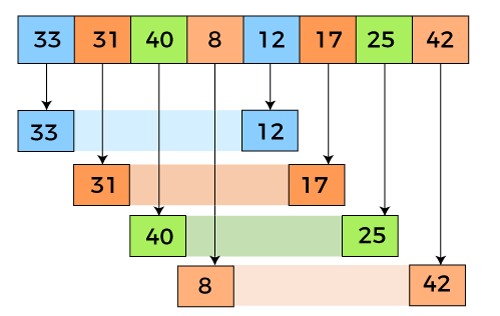
## 1.3. Thuật toán sắp xếp Shell Sort

### 1.3.1. Ý tưởng chính của thuật toán

Thuật toán Shell Sort, còn gọi là sắp xếp giảm dần khoảng cách, là một cải tiến của thuật toán sắp xếp chèn (Insertion Sort). Ý tưởng chính của Shell Sort là sắp xếp các phần tử ở khoảng cách xa nhau trước, sau đó dần dần giảm khoảng cách giữa các phần tử được so sánh. Bằng cách này, khi khoảng cách giảm xuống còn rất nhỏ, các phần tử gần như đã ở trên đúng vị trí, giúp thuật toán chèn trở nên hiệu quả hơn.Ví dụ minh hoạ:

Cho dãy số a: 33, 31, 40, 8, 12, 17, 25, 42. Sắp xếp dãy số này theo thứ tự tăng dần bằng thuật toán Shell Sort được minh họa như sau:

Hình 1.1 Minh hoạ từng bước thuật toán Shell Sort



### 1.3.2 Lưu đồ thuật toán

1. **Khởi Tạo Mảng**: Xác định mảng cần sắp xếp.
2. **Chọn Gap**: Chọn khoảng cách (gap) ban đầu thường là n/2, với n là số phần tử trong mảng.
3. **So Sánh và Đổi vị trí**: Trong mỗi nhóm phần tử cách nhau một khoảng gap, so sánh 2 phần tử và đổi chỗ vị trí phần tử nếu cần thiết
4. **Giảm Gap**: Chia gap đã chọn cho 2 để có gap mới.
5. **Lặp Lại**: Quay lại bước 4 nếu gap vẫn lớn hơn 1.
6. **Kết Thúc**: Khi gap = 1, thuật toán sẽ hoàn thành việc sắp xếp, đánh dấu điểm kết thúc của thuật toán.

### 1.3.3 Đánh giá thuật toán

#### 1.3.3.1 Độ phức tạp

Thuật toán Shell Sort không có một công thức độ phức tạp vì độ phức tạp của nó phụ thuộc vào khoảng cách (gap) giữa các phần tử khi sắp xếp, và có nhiều cách khác nhau để chọn khoảng cách này.

* **Trường hợp tốt**: Khi dãy đã gần sắp xếp hoặc khi các khoảng cách được chọn tối ưu, độ phức tạp có thể đạt đến O(nlog²n).
* **Trung bình**: Độ phức tạp trung bình của Shell Sort thường được coi là khoảng O(n(logn)²), tuy nhiên, điều này có thể thay đổi tùy thuộc vào cách chọn các khoảng cách.
* **Trường hợp xấu**: Trong trường hợp xấu nhất, độ phức tạp có thể lên đến O(n²), đặc biệt khi các khoảng cách không được chọn một cách tối ưu.

**Không Gian Bộ Nhớ Sử Dụng**: O(1)

* Shell Sort là một thuật toán sắp xếp tại chỗ, không cần thêm không gian lưu trữ ngoài mảng đang sắp xếp.

**Mô Hình Thuật Toán**: Tăng Dần Khoảng Cách

* Shell Sort làm việc bằng cách giảm dần khoảng cách giữa các phần tử được so sánh và đổi chỗ, qua mỗi lần lặp, nó không theo mô hình chia và chinh phục.

**Ổn Định**: Không

* Shell Sort không phải là thuật toán ổn định vì nó có thể thay đổi thứ tự của các phần tử có giá trị bằng nhau trong mảng ban đầu.

#### Ưu điểm và nhược điểm

Ưu điểm:

Hiệu suất tốt hơn: Shell Sort có hiệu suất tốt hơn so với một số thuật toán sắp xếp đơn giản khác như Insertion Sort hay Bubble Sort. Đặc biệt, khi sử dụng các khoảng cách gap phù hợp, Shell Sort có thể giảm thiểu số lần so sánh và di chuyển dữ liệu, cải thiện tốc độ sắp xếp.

Dễ cài đặt và sử dụng: Thuật toán Shell Sort không yêu cầu kiến thức phức tạp và có cài đặt đơn giản. Nó có thể dễ dàng được sử dụng trong các ứng dụng thực tế.

Khả năng sắp xếp được các dạng dữ liệu khác nhau: Shell Sort có khả năng sắp xếp được các dạng dữ liệu khác nhau, bao gồm cả dữ liệu đã sắp xếp gần như hoàn chỉnh và dữ liệu ngẫu nhiên.

Nhược điểm:

Độ phức tạp không đảm bảo: Độ phức tạp của Shell Sort phụ thuộc vào cách chọn khoảng cách gap và cách thực hiện việc sắp xếp các nhóm con. Trong trường hợp xấu nhất, thuật toán có thể có độ phức tạp là O(n^2), gây tốn thời gian xử lý đối với dữ liệu lớn và không gần như đã sắp xếp.

Không ổn định: Shell Sort không đảm bảo tính ổn định trong việc sắp xếp các phần tử có giá trị bằng nhau. Điều này có nghĩa là thứ tự ban đầu của các phần tử có thể bị thay đổi sau khi sắp xếp.

#### 1.3.3.3 Tốc độ thực hiện

Tốc độ thực hiện của một thuật toán sắp xếp như Shell Sort phụ thuộc vào nhiều yếu tố như:

**Kích Thước Dữ Liệu**: Lượng phần tử cần sắp xếp (n) ảnh hưởng trực tiếp đến tốc độ thực hiện. Các thuật toán có độ phức tạp thời gian thấp hơn sẽ thực hiện nhanh hơn trên dữ liệu lớn.

**Trạng Thái Ban Đầu của Dữ Liệu**: Nếu dữ liệu đã gần được sắp xếp từ trước, Shell Sort có thể thực hiện nhanh hơn so với trường hợp dữ liệu ban đầu hoàn toàn ngẫu nhiên hoặc trong trạng thái xấu nhất (ví dụ: hoàn toàn sắp xếp ngược).

**Lựa Chọn Khoảng Cách (Gap Sequence)**: Cách chọn khoảng cách giữa các phần tử để áp dụng sắp xếp chèn có ảnh hưởng lớn đến hiệu suất. Có nhiều dãy khoảng cách được đề xuất như dãy của Shell, dãy của Knuth, v.v., và mỗi dãy sẽ có hiệu suất khác nhau tùy thuộc vào dữ liệu đầu vào.

**Phần Cứng và Môi Trường Thực Thi**: Tốc độ xử lý của CPU, loại bộ nhớ (RAM, SSD, HDD), và sự tối ưu hóa của trình biên dịch hoặc ngôn ngữ lập trình cũng là các yếu tố quan trọng.

## 1.4. Dev C++

Bloodshed Dev-C ++ (https://www.bloodshed.net/devcpp.html) là môi trường phát triển tích hợp (IDE) đầy đủ tính năng cho ngôn ngữ lập trình C/C ++ sử dụng Mingw của GCC (Bộ sưu tập trình biên dịch GNU) làm trình biên dịch.

Dev-C ++ cũng có thể kết hợp với Cygwin hoặc bất kỳ trình biên dịch dựa trên GCC nào khác. Các tính năng của Dev-C++:

* Hỗ trợ trình biên dịch dựa trên GCC.
* Gỡ lỗi tích hợp (sử dụng GDB- General DeBug).
* Quản lý dự án.
* Trình chỉnh sửa cú pháp.
* Trình duyệt lớp.
* Hoàn thành mã.
* Danh sách chức năng.
* Hồ sơ hỗ trợ.
* Nhanh chóng tạo Windows, console, thư viện tĩnh và DLL13.
* Hỗ trợ các mẫu để tạo các loại dự án của riêng bạn.
* Tạo Makefile.
* Chỉnh sửa và biên dịch các tệp Tài nguyên.
* Quản lý công cụ.
* Hỗ trợ in.
* Tìm và thay thế mã lệnh.
* Hỗ trợ CVS.

## 1.5. Thư viện Graphics.h

Vì sử dụng Dev C++ làm trình biên dịch cho việc cài đặt thuật toán nên không thể

thực hiện trên môi trường Windows. Vì vậy, một môi trường giả lập graphic của Borland C được Michael tạo ra thư viên có tên là Graphics.h. để có thể làm được điều đó. Micheal đã thay đổi BGI library (thư viện BGI) thành thư viện có tên WinBGIm để có thể sử dụng tốt trên windows. Và bây giờ bạn đã có thể sử dụng tốt các hàm đặc biệt của borland bằng Dev C++ (https://github.com/SagarGaniga/Graphics-Library)

# CHƯƠNG 2. PHƯƠNG PHÁP NGHIÊN CỨU

## 2.1. Cài đặt Dev C và thư viện graphics.h

Tải file cài đặt phần mềm Dev C++ theo đường dẫn trong mục 1.4. Sau đó mở file

vừa tải, và tiến hành cài đặt. Thư viện graphics.h được tiến hành cài đặt theo các bước:

**Bước 1:** Copy 6-ConsoleAppGraphics và ConsoleApp\_cpp\_graph

Paste C:\...\Dev-Cpp\Templates

**Bước 2:** Copy graphics và winbgim

Paste C:\... Dev-Cpp \MinGW64\x86\_64-w64-mingw32\include

**Bước 3:** Copy libbgi.a

Paste C:\...\Dev-Cpp\MinGW64\x86\_64-w64-mingw32\lib

**Bước 4:** Ở DevC++ => New Project => Console Graphics Application

**Bước 5:** Thay đổi Tools - Compiler Option: TDM – GCC 4.9.2 32 bit Release

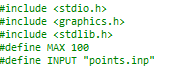
**Bước 6:** Sử dụng đoạn code mẫu bên dưới để test thư viện winbgim

|  |  |
| --- | --- |
|  |  |

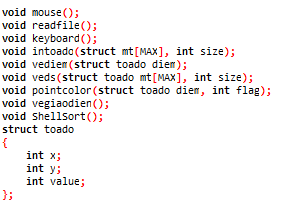
Hình 2.1. Ví dụ minh hoạ thư viện graphics.h

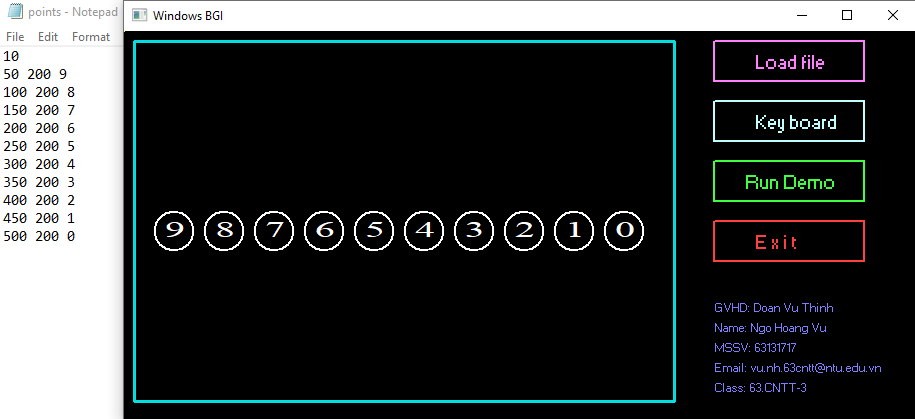
## 2.2. Cài đặt thuật toán Shell Sort

**Khai báo thư viện**



Khai báo hàm và biến



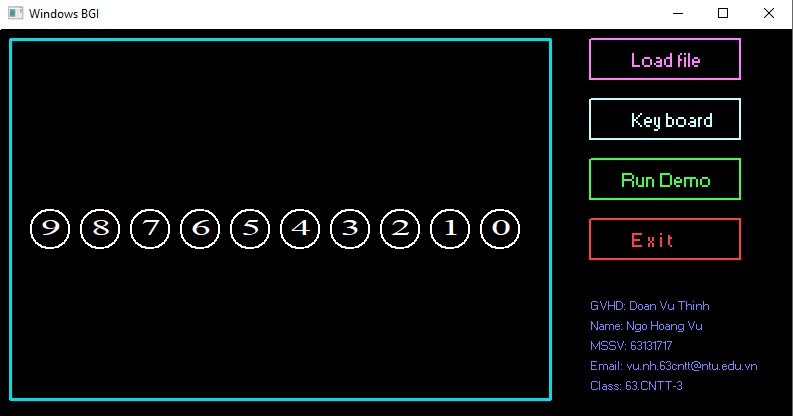


Hình 2.1. Màn hình lấy dữ liệu từ file

Hiển thị lên màn hình đồ hoạ

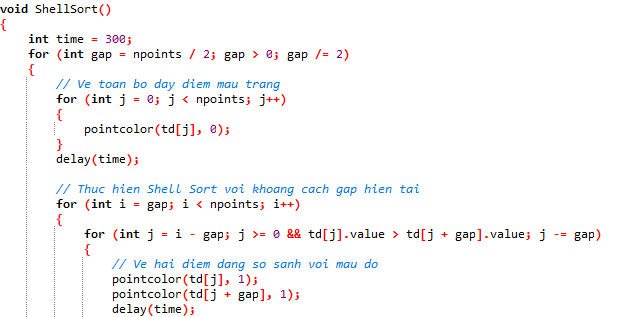
|  |
| --- |
|  |
|  |

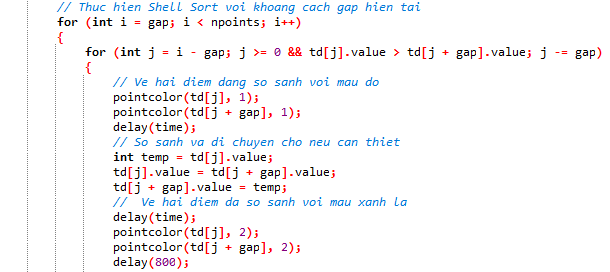
Hình 2.2. Code hiển thị giao diện

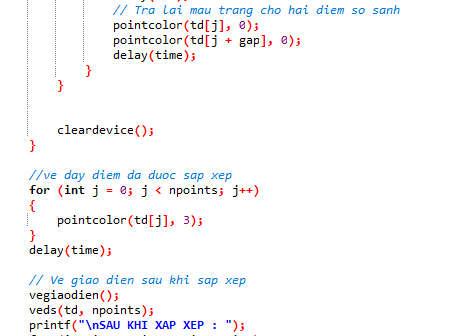
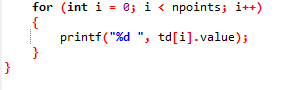


Hình 2.3. Màn hình hiển thị các phần tử

Thuật toán Shell Sort:



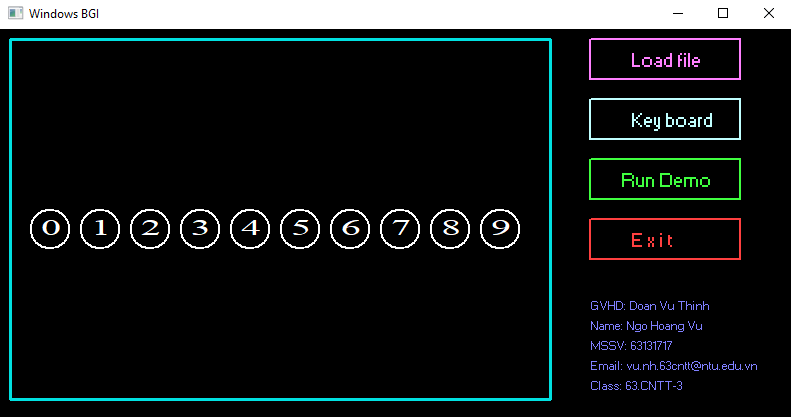


Hình 2.7. Code Shell Sort

# CHƯƠNG 3. KẾT QUẢ THỰC HIỆN

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Hình 3.1. Video Shell Sort thực hiện sắp xếp

Hình 3.2. Merge Sort hoàn thành việc sắp xếp phần tử

# CHƯƠNG 4: THẢO LUẬN

Đề tài đã mô phỏng quá trình sắp xếp mảng bằng thuật toán Shell Sort . Ngoài ra còn tìm hiểu thêm về thuật toán Bubble Sort và Quick Sort, thấy được thuật toán đã có cải tiến hơn so với Bubble, Quick Sort.

Đề tài vẫn còn nhiều thiếu xót và lỗi. Trong tương lai, em sẽ tiếp tục tìm hiểu thêm và sẽ cố gắng tìm hiểu và cải tiến thuật toán Shell Sort.

# TÀI LIỆU THAM KHẢO

1. “Thuật toán sắp xếp”, [Wikipedia Tiếng Việt](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n_s%E1%BA%AFp_x%E1%BA%BFp)