

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



Niên luận Cơ sở ngành Kỹ thuật phần mềm
TRÌNH BÀY GIẢI THUẬT SẮP XẾP
COCKTAIL

Sinh viên thực hiện:
Họ và tên: Kiều Tấn Quốc
MSSV: B2012132

Giảng viên hướng dẫn:
PGS.TS. Huỳnh Xuân Hiệp
MSCB: 001067

Học kỳ 2, năm học 2022 - 2023

ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN NIÊN LUẬN CƠ SỞ NGÀNH KTPM
(Học kỳ: 02, năm học 2022 – 2023)

TÊN ĐỀ TÀI: TRÌNH BÀY GIẢI THUẬT SẮP XẾP COCKTAIL

GIẢNG VIÊN HƯỚNG DẪN:

STT	HỌ VÀ TÊN	MSCB
1	TS. Huỳnh Xuân Hiệp	001067

SINH VIÊN THỰC HIỆN:

HỌ VÀ TÊN	MSSV	THƯỜNG (Tối đa 1 điểm)	ĐIỂM (Thang điểm 10)
Kiều Tấn Quốc	B2012132		

I. HÌNH THỨC (0.5 điểm)

Bìa (tối đa 0.25 điểm)

- Đầy đủ các thông tin
- Đúng định dạng

Bố cục (Tối đa 0.25 điểm)

- Trang đánh giá kết quả thực hiện niên luận
- Mục lục: cấu trúc chương, mục, tiểu mục
- Phụ lục (nếu có)
- Tài liệu tham khảo

II. NỘI DUNG (3.5 điểm)

Giới thiệu (Tối đa 0.5 điểm)

- Mô tả bài toán, mục tiêu cần đạt được (0.25 điểm)
- Hướng giải quyết và kế hoạch thực hiện (0.25 điểm)

Lý thuyết (tối đa 0.5 điểm)

- Các khái niệm sử dụng trong đề tài (0.25 điểm)
- Kết quả vận dụng lý thuyết vào đề tài (0.25 điểm)

Ứng dụng (tối đa 2.0 điểm)

- Phân tích yêu cầu bài toán, xây dựng các cấu trúc dữ liệu cần thiết (0.5 điểm)
- Sơ đồ chức năng, lưu đồ giải thuật giải quyết vấn đề (1.0 điểm)
- Giới thiệu chương trình (0.5 điểm)

Kết luận (tối đa 0.5 điểm)

- Nhận xét kết quả đạt được

III. CHƯƠNG TRÌNH DEMO (5.0 điểm)	
Giao diện thân thiện với người dùng (1.0 điểm)	
Hướng dẫn sử dụng (0.5 điểm)	
Kết quả thực hiện đúng với kết quả của phần ứng dụng (tối đa 3.5 điểm)	
<ul style="list-style-type: none"> ▪ Kết quả đúng (2.0 điểm) ▪ Cách thức thực hiện hợp lý (1.0 điểm) ▪ Chức năng bổ sung, sáng tạo (0.5 điểm) 	

Ghi chú: Điểm trong khung “các sinh viên thực hiện” là điểm kết quả cuối cùng của từng sinh viên trong quá trình thực hiện niên luận.

Nếu sinh viên demo chương trình và trả lời vấn đáp không đạt yêu cầu của giảng viên hướng dẫn thì sinh viên sẽ nhận điểm F cho học phần này.

Cần Thơ, ngày tháng năm 2023

GIẢNG VIÊN CHẤM

TS. Huỳnh Xuân Hiệp

LỜI CẢM ƠN

Sau một thời gian tiến hành triển khai nghiên cứu, em cũng đã hoàn thành nội dung niên luận cơ sở ngành Kỹ thuật phần mềm. Niên luận được hoàn thành không chỉ là công sức của bản thân em mà còn có sự giúp đỡ, hỗ trợ rất nhiều từ quý Thầy, cô của Trường Công nghệ Thông tin và Truyền thông và bạn bè.

Trước hết, em xin gửi lời cảm ơn chân thành và sâu sắc đến TS. Huỳnh Xuân Hiệp, người trực tiếp hướng dẫn niên luận cho em. Thầy đã dành nhiều thời gian, tâm sức, cho em nhiều ý kiến, nhận xét quý báu, chỉnh sửa cho em những chi tiết nhỏ trong đề tài niên luận, giúp niên luận của em được hoàn thiện hơn về mặt nội dung và hình thức. Thầy đã luôn quan tâm, động viên, nhắc nhở kịp thời để em có thể hoàn thành niên luận đúng tiến độ.

Em cũng xin gửi lời cảm ơn đến bạn bè, anh/chị khóa trên cũng đã có những ý kiến đóng góp vô cùng quý báu, động viên và chỉ dẫn tận tình trong suốt quá trình em thực hiện niên luận, vừa giúp em có được sự khích lệ, tin tưởng vào bản thân em, vừa tạo động lực nhắc nhở em có trách nhiệm với đề tài của mình, giúp em hoàn chỉnh niên luận tốt hơn.

Cuối cùng, em xin kính chúc quý Thầy, cô, bạn bè thật dồi dào sức khỏe, ngày càng thành công trong cuộc sống và công việc!

Cần Thơ, ngày tháng năm 2023

Sinh viên thực hiện

Kiều Tấn Quốc

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN	1
I. MÔ TẢ BÀI TOÁN.....	1
II. MỤC TIÊU CẦN ĐẠT ĐƯỢC.....	2
III. HƯỚNG GIẢI QUYẾT	2
IV. KẾ HOẠCH THỰC HIỆN.....	3
CHƯƠNG 2. LÝ THUYẾT	3
I. CÁC KHÁI NIỆM	3
2.1.1. Khái niệm sắp xếp, thuật toán sắp xếp và xử lý bất đồng bộ.	3
2.1.2. Thuật toán sắp xếp Cocktail	3
II. KẾT QUẢ VẬN DỤNG LÝ THUYẾT TRONG ĐỀ TÀI	4
CHƯƠNG 3. KẾT QUẢ ỨNG DỤNG.....	4
I. PHÂN TÍCH YÊU CẦU BÀI TOÁN	4
II. XÂY DỰNG CẤU TRÚC DỮ LIỆU CẦN THIẾT	5
3.2.1. Hàm tạo dữ liệu sắp xếp	5
3.2.2. Hàm nhập dữ liệu từ tập tin	7
3.2.3. Hàm thuật toán sắp xếp Cocktail.....	8
3.2.4. Hàm khởi tạo các ô nhập phần tử	14
3.2.5. Hàm tạo dữ liệu từ bàn phím	15
3.2.6. Hàm quyết định tạo tự động hay từ bàn phím người dùng.....	16
3.2.7. Hàm xóa tập tin hiện tại.....	16
3.2.8. Hàm tạm dừng hoặc tiếp tục thuật toán	17
3.2.9. Hàm ràng buộc ô “Số lượng”	17
III. THIẾT KẾ GIẢI THUẬT (LƯU ĐỒ - NGÔN NGỮ GIẢ)	18
IV. GIỚI THIỆU TRANG WEB.....	19
3.4.1. Giao diện chính.....	19
3.4.2. Giao diện nhập dữ liệu từ bàn phím	21
3.4.3. Giao diện dữ liệu được hiển thị thông qua nhập từ bàn phím	21
CHƯƠNG 4. KẾT LUẬN – ĐÁNH GIÁ.....	22
I. KẾT QUẢ ĐẠT ĐƯỢC.....	22
4.1.1. Về mặt chuyên môn	22

4.1.2. Về mặt kinh nghiệm.....	22
--------------------------------	----

CHƯƠNG 1. TỔNG QUAN

I. MÔ TẢ BÀI TOÁN

Hiện nay, hoạt động tìm kiếm đã góp mặt ở khắp mọi nơi trong đời sống (tìm sách trong thư viện, tìm kiếm một món hàng, tra từ điển, tìm kiếm các quán ăn gần nhà,...). Tuy nhiên, để đảm bảo được khả năng tìm kiếm một cách trơn tru và ít gặp trở ngại, nhanh chóng và chính xác thì bắt buộc dữ liệu đang tìm kiếm cần được tổ chức, sắp xếp một cách hợp lý và có khoa học, ngăn nắp theo một trật tự nhất định. Khi đó, việc tìm kiếm sẽ dễ dàng và linh hoạt hơn. Điều này có ý nghĩa vô cùng quan trọng trong quá trình quản lý và lưu trữ dữ liệu.

Tuy việc sắp xếp thông tin là cần thiết, nhưng chúng ta lại phải đối mặt với việc trên thế giới đã cho ra đời rất nhiều phương pháp sắp xếp khác nhau, bên cạnh việc giúp mọi người có thể linh hoạt và sáng tạo hơn trong việc sắp xếp dữ liệu của mình, thì còn gây khó khăn trong việc tìm hiểu, hình dung và lựa chọn một thuật toán phù hợp. Mỗi thuật toán sắp xếp đều có ưu điểm và nhược điểm khác nhau tùy vào mục đích sử dụng (giảng dạy trong nhà trường, ứng dụng trong thực tế,...). Trong khuôn khổ giảng dạy trong nhà trường, làm thế nào để người dùng có thể hình dung ra cách thức hoạt động của một phương pháp sắp xếp một cách nhanh nhất, để có thể đưa ra lựa chọn có nên sử dụng phương pháp này hay không?

Nắm bắt được những nhu cầu thiết yếu đó, trang web “Mô tả thuật toán sắp xếp Cocktail” đã ra đời. Trang web giúp người dùng có cái nhìn cụ thể nhất về cách thức hoạt động của thuật toán sắp xếp Cocktail một cách dễ hiểu nhất. Qua đó, phần mềm giúp ta có thể thấy được với mỗi dòng mã lệnh tương ứng với những công việc nào và sẽ hoạt động ra sao.

Trang web được hình thành với những yếu tố hình ảnh trực quan, nội dung xúc tích, sinh động nhằm biểu diễn cách thức hoạt động của thuật toán một cách dễ hiểu nhất mà không cần phải cố gắng quá nhiều trong việc tìm hiểu mã lệnh. Từ tất cả những yếu tố trên, trang web “Mô tả thuật toán sắp xếp Cocktail” hy vọng có thể giúp đỡ sinh viên trong việc học Cấu trúc dữ liệu và giải thuật trở nên dễ dàng mà không gây nhàm chán bởi những dòng mã lệnh khô khan.

II. MỤC TIÊU CẦN ĐẠT ĐƯỢC

- Thể hiện được các quá trình mà thuật toán hoạt động bằng giao diện đồ họa.
- Thể hiện được việc sắp xếp của thuật toán trên từng dòng mã lệnh diễn ra như thế nào.
- Biết cách đưa trang web lên host để triển khai đến nhiều người dùng khác nhau.
- Biết cách sử dụng hàm xử lý bất đồng bộ (Asynchronous) trong Javascript.
- Nắm vững được kiến thức về cấu trúc dữ liệu và giải thuật về thuật toán được sắp xếp tại trang web.
- Có thêm kiến thức về các thư viện bổ sung trong Javascript.

III. HƯỚNG GIẢI QUYẾT

Trước tiên, cần tìm hiểu kỹ cách thức hoạt động của thuật toán Cocktail. Sau khi xác định được thuật toán là biến thể của thuật toán sắp xếp nổi bọt (Bubble Sort), cần phải xem lại cách thức hoạt động cụ thể của thuật toán sắp xếp nổi bọt và tìm ra điểm khác biệt trong việc triển khai.

Sau khi tìm hiểu kỹ và tìm ra sự khác biệt, tiến hành vào bước triển khai bằng giao diện đồ họa. Dữ liệu sắp xếp có thể được khởi tạo bằng 3 cách: tạo ngẫu nhiên (khi người dùng không nhập số phần tử vào ô nhập hoặc không có tập tin nào được chọn thì trang web sẽ tự tạo ra dãy giá trị ngẫu nhiên), nhập giá trị từ bàn phím (cho phép người dùng nhập số lượng phần tử vào ô nhập và lần lượt nhập giá trị cho từng phần tử đó), nhập từ tập tin (cho phép người dùng nhập các giá trị sắp xếp thông qua một tập tin văn bản chứa số lượng phần tử và các phần tử có giá trị tương ứng và dữ liệu sẽ được tạo ra ngay khi tập tin được đưa vào trang web). Trong trường hợp tập tin văn bản có nội dung không hợp lệ (số phần tử khác với số lượng phần tử hoặc ngược lại) hệ thống sẽ báo lỗi và yêu cầu người dùng nhập lại tập tin. Bên cạnh đó còn có các chức năng như: Tiến 1 bước, lùi 1 bước, tạm dừng, tiếp tục, xóa dữ liệu hiện có và hiện ra code minh họa. Người dùng chỉ cần bấm vào nút tương ứng để được thực hiện các chức năng tương ứng. Trang web sẽ hiện các thông báo kèm theo để người dùng hiểu rõ hơn trong lúc sử dụng.

Tiếp theo, sau khi người dùng chạy thuật toán, cần phải có hiện khung code minh họa để người dùng theo dõi (có thể tắt/mở nếu thấy không cần thiết). Người dùng có thể tạm dừng, tiến 1 bước hoặc lùi 1 bước.

Cuối cùng, khi thuật toán đã sắp xếp hoàn tất, người dùng có nhu cầu tạo dữ liệu mới chỉ cần bấm nút xóa, trang web sẽ xóa dữ liệu nhập tay hiện có (hoặc tập tin hiện có) để người dùng có thể nhập lại dữ liệu mới khi cần.

IV. KẾ HOẠCH THỰC HIỆN

Tuần	Công việc
1	Nghiên cứu đề tài
2-5	Phác thảo và cài đặt các cấu trúc dữ liệu và chức năng ban đầu của trang web.
6-8	Đổi từ dạng so sánh cột sang so sánh thực tế hơn, giao diện trực quan hơn. Tuỳ chỉnh và thêm các chức năng mới.
9-10	Tinh chỉnh Javascript
10-12	Viết báo cáo và kết thúc

CHƯƠNG 2. LÝ THUYẾT

I. CÁC KHÁI NIỆM

2.1.1. Khái niệm sắp xếp, thuật toán sắp xếp và xử lý bất đồng bộ.

Sắp xếp là quá trình bố trí lại các phần tử trong một tập hợp theo một trình tự nào đó nhằm mục đích giúp quản lý và tìm kiếm các phần tử dễ dàng và nhanh chóng hơn.

Thuật toán sắp xếp là một thuật toán sắp xếp các phần tử của một danh sách (hoặc một mảng) theo thứ tự (tăng hoặc giảm). Và để dễ dàng cho quá trình nghiên cứu và học tập thì người ta thường gán các bằng phần tử bởi một giá trị số nguyên.

Xử lý bất đồng bộ (Asynchronous) là cách mà nhiều công việc có thể thực hiện cùng một lúc. Và nếu công việc thứ hai kết thúc trước, nó có thể sẽ cho ra kết quả trước cả câu lệnh thứ nhất. Vì thế, đôi khi kết quả của các câu lệnh sẽ không trả về theo đúng thứ tự như trên dòng code.

2.1.2. Thuật toán sắp xếp Cocktail

Khái niệm: là một cải tiến của Bubble Sort. Sau khi đưa phần tử nhỏ nhất về đầu dãy, thuật toán sẽ giúp chúng ta đưa phần tử lớn nhất về cuối dãy. Do đưa các phần tử về đúng vị trí ở cả hai đầu nên thuật toán sắp xếp Cocktail sẽ giúp cải thiện thời gian sắp xếp dãy số.

Ý nghĩa tên gọi: tên của thuật toán này được đặt theo cách hoạt động của nó là “làm nóng và lắng”, tương tự như cách pha chế món Cocktail. Khi pha chế một món Cocktail, người pha chế sẽ cho các nguyên liệu vào trong một chiếc shaker và lắc đều để các thành phần hòa quyện với nhau. Tương tự, thuật toán hoạt động bằng cách “làm nóng” danh sách từ đầu tới cuối và “lắc” danh sách từ cuối lên đầu, để đưa phần tử lớn nhất về cuối dãy và phần tử nhỏ nhất về đầu dãy. Lặp đi lặp lại tới khi việc sắp xếp được hoàn thành.

II. KẾT QUẢ VẬN DỤNG LÝ THUYẾT TRONG ĐỀ TÀI

Áp dụng thuật toán sắp xếp Cocktail để đưa vào trang web có giao diện đồ họa. Sử dụng css transition để giúp người dùng thấy được sự hoán đổi giá trị của hai phần tử. Dùng hàm bất đồng bộ để tránh hiện tượng chặn luồng thực thi khi đợi kết quả của một hành động khác, ngoài ra nó còn giúp tăng hiệu suất và đơn giản hóa mã nguồn.

CHƯƠNG 3. KẾT QUẢ ỨNG DỤNG

I. PHÂN TÍCH YÊU CẦU BÀI TOÁN

Trang web “mô tả thuật toán sắp xếp Cocktail” với mục tiêu trọng tâm và thiết yếu là phải mô tả được cách thức hoạt động của thuật toán một cách trực quan, dễ hiểu thông qua giao diện đồ họa.

Dữ liệu đầu vào của trang web sẽ gồm 1 tập hợp các phần tử có giá trị ngẫu nhiên, có thể được tạo ngẫu nhiên, nhập từ bàn phím hoặc nhập từ tập tin tùy vào mục đích của người dùng lựa chọn. Cụ thể, đầu tiên người dùng sẽ nhập vào số lượng phần tử cần sắp xếp. Sau đó, hệ thống sẽ có 3 lựa chọn để người dùng có thể tạo ra dữ liệu sắp xếp bao gồm: tạo ngẫu nhiên, nhập từ bàn phím và nhập từ tập tin.

Đối với tạo ngẫu nhiên: Ở phương pháp này, người dùng chỉ cần click vào nút “Tạo ngẫu nhiên”, chương trình sẽ tạo ra dãy số bằng cách chọn ra một số ngẫu nhiên cho từng giá trị tương ứng với số lượng phần tử mà người dùng đã nhập từ trước (trong trường hợp người dùng không nhập số lượng phần tử trang web sẽ mặc định tạo ra dãy số gồm 20 phần tử). Để thực hiện việc này, ta cần thiết kế một hàm có tên là `initialBars()` để truyền giá trị bằng vòng lặp. Kèm theo đó là lệnh rẽ nhánh `if` trong trường hợp người dùng không nhập số phần tử đầu vào hoặc nhập số phần tử nhưng không nhập giá trị cho từng phần tử thì trang web sẽ tự động tạo ra dãy số ngẫu nhiên gồm 20 phần tử. Còn khi người dùng đã nhập đầy đủ thì hệ thống sẽ tạo ra dữ liệu dựa vào dữ liệu người dùng nhập vào.

Đối với nhập từ bàn phím: Ở phương pháp này, ta sẽ thiết kế một hàm mang tên `createInput()` để hiện ra các ô cho phép người dùng nhập vào các phần tử tương ứng. Sau khi có số lượng phần tử, ta tiến hành thực hiện vòng lặp để khởi tạo các thanh có chiều cao và giá trị của các phần tử tương ứng. Mỗi hàng sẽ gồm 24 ô nhập để người dùng thực hiện nhập giá trị.

Đối với nhập từ tập tin: Cách này sẽ giúp người dùng nhập được nhanh hơn các giá trị của các phần tử chỉ bằng một tập tin văn bản. Với cách này, người dùng sẽ thực hiện ít thao tác hơn, từ đó hiệu suất của trang web sẽ tăng lên, hiệu quả hơn. Để bắt đầu cho việc sử dụng tập tin, ta cần bắt sự kiện `change` cho ô input có kiểu là `file`. Khi giá trị trong ô input có sự thay đổi (cụ thể là có 1 tập tin mới được thêm vào). Sau khi bắt được sự kiện đó, Javascript sẽ tiến hành đọc nội dung tập tin và gán giá trị vào 1 biến. Sau đó, tiến hành

đọc hàng đầu tiên (số lượng phần tử) và dùng kỹ thuật cắt một chuỗi lớn thành 1 mảng nhờ vào phương thức `split()` của String. Khi tách ra thành các giá trị con, ta sử dụng vòng lặp để tiến hành khởi tạo dữ liệu so sánh dựa trên giá trị con đã được cắt từ một chuỗi lớn. Người dùng sau khi nhập được tập tin vào trang web, trang web sẽ tự động tạo ra các dữ liệu so sánh mà không cần phải ấn thêm bất kỳ nút chức năng nào khác. Cuối cùng, người dùng thực hiện các chức năng so sánh tương tự như 2 cách nhập trên.

Trang web cho phép người dùng tạm dừng hoặc tiếp tục thuật toán bất kỳ lúc nào. Ngoài ra, khi cần thiết người dùng có thể ấn vào nút hiện code để xem toàn bộ quá trình thực hiện của thuật toán.

Kết quả cuối cùng của trang web, sẽ là một dãy dữ liệu bao gồm các số và các cột đã được sắp xếp theo thứ tự tăng dần cùng với các cột được hiển thị bằng màu xanh lá nhằm biểu thị rằng thuật toán đã hoàn thành để người dùng có thể kiểm tra lại kết quả.

II. XÂY DỰNG CẤU TRÚC DỮ LIỆU CẦN THIẾT

Để thực hiện cho việc mô phỏng thuật toán sắp xếp, ta cần cài đặt cấu trúc dữ liệu, các hàm/thủ tục cần thiết như sau:

3.2.1. Hàm tạo dữ liệu sắp xếp

```
const container = document.querySelector(".data-container");
function initialBars() {
  const myElement = document.getElementById("space_sort");
  myElement.innerHTML = "";
  let number = document.getElementById("quantity").value;
  if (number == 0) {
    let num = 15;
    for (let i = 0; i < num; i += 1) {
      let value = Math.floor(Math.random() * 100) + 1;
      let bar = document.createElement("div");
      bar.classList.add("bar");
      bar.style.transform = `translateX(${i * 40}px)
      translateY(${Math.floor(Math.random() * 80) - 40}px)
      rotate(${Math.floor(Math.random() * 121) - 60}deg)`;
      let barLabel = document.createElement("label");
      barLabel.classList.add("bar__id");
      barLabel.innerHTML = value;
      bar.appendChild(barLabel);
      myElement.appendChild(bar);
    }
  }
}
```

```

if (number != 0) {
    let number = document.getElementById("quantity").value;
    for (let i = 0; i < number; i += 1) {
        let value = Math.floor(Math.random() * 100) + 1;
        let bar = document.createElement("div");
        bar.classList.add("bar");
        bar.style.transform = `translateX(${i * 40}px)
translateY(${Math.floor(Math.random() * 80) - 40}px)
rotate(${Math.floor(Math.random() * 121) - 60}deg)`;
        let barLabel = document.createElement("label");
        barLabel.classList.add("bar__id");
        barLabel.innerHTML = value;
        bar.appendChild(barLabel);
        myElement.appendChild(bar);
    }
}
}

```

Để tạo hàm dữ liệu sắp xếp, ta dùng phương thức `querySelector()` và DOM để lấy được vị trí của thẻ sẽ chứa dữ liệu sắp xếp và giá trị của ô nhập “số lượng phần tử” do người dùng nhập vào. Hàm sẽ dùng 2 lệnh rẽ nhánh, tương ứng với 2 điều kiện: Nếu người dùng không nhập số lượng phần tử thì website sẽ thực hiện các câu lệnh để tạo ra dãy số ngẫu nhiên có 20 phần tử, nếu người dùng nhập vào một số bất kỳ, hệ thống sẽ tạo ra dãy số có số lượng phần tử tương ứng thông qua vòng lặp.

3.2.2. Hàm nhập dữ liệu từ tập tin

```
const container = document.querySelector(".data-container");

let upload = document.getElementById('uploadFile');
let output = document.getElementById('output');
let quantity, temp;

uploadFile.addEventListener('change', () => {
    let fr = new FileReader();
    fr.readAsText(uploadFile.files[0]);
    fr.onload = function () {
        let text = fr.result;
        quantity = text.split('\n').shift(); // Nhận hàng đầu tiên là số phần tử
        cần sắp xếp
        let txt = text.split('\n');
        let substring = txt[1].split(" ");
        // console.log(substring);
        temp = parseInt(quantity);
        if (temp == substring.length) {
            const myElement = document.getElementById("container");
            myElement.innerHTML = '';
            for (let i = 0; i < substring.length; i++) {
                let value = substring[i];
                let bar = document.createElement("div");
                bar.classList.add("bar");
                bar.style.height = `${value * 3}px`;
                bar.style.transform = `translateX(${i * 30}px)`;
                let barLabel = document.createElement("label");
                barLabel.classList.add("bar__id");
                barLabel.innerHTML = value;
                bar.appendChild(barLabel);
                container.appendChild(bar);
            }
        } else {
            alert("Số phần tử không khớp với số lượng phần tử đã thiết lập! Vui lòng kiểm tra lại tập tin");
        }
        if (temp != document.getElementById('quantity').value) {
            document.getElementById('quantity').value = temp;
        }
        // if (temp != substring.length) {
        //     alert("Số lượng phần tử phải là số dương! Vui lòng kiểm tra lại tập tin");
        // }
    }
});
```



```

bars[i].style.backgroundImage = "linear-gradient(90deg, pink, red)";
bars[i + 1].style.backgroundImage = "linear-gradient(90deg, pink, red)";
if (value1 > value2) {
    document.getElementById("messageDes").innerHTML = "If " + value1 + " > " +
value2;

    var temp2 = bars[i].childNodes[0].innerText;
    await new Promise((resolve) => {
        setTimeout(() => {
            resolve();
        }, timeoutValue)
    });
    document.getElementById("messageDes").innerHTML = "Swap " + value1 + " and " +
value2;

    bars[i].style.transform = `translateX(${i * 40}px) translateY(${transY2}px)
rotate(${rotate2}deg)`;
bars[i + 1].style.transform = `translateX(${(i + 1) * 40}px)
translateY(${transY1}px) rotate(${rotate1}deg)`;
bars[i].childNodes[0].innerText = bars[i + 1].childNodes[0].innerText;
bars[i + 1].childNodes[0].innerText = temp2;
// Đặt lại giá trị swapped là true nếu có hoán đổi
swapped = true;
}else{
    document.getElementById("messageDes").innerHTML = "If " + value1 + " < " +
value2 + " => " + "do nothing" ;
}

    await new Promise((resolve) => {
        setTimeout(() => {
            resolve();
        }, timeoutValue)
    });

```

```

bars[i].style.backgroundImage = "linear-gradient(90deg, #BFEFFF,#6FB3FF)";
    bars[i + 1].style.backgroundImage = "linear-gradient(90deg, #BFEFFF,#6FB3FF)";
    await waitforme(100);
    if (flag == 1) await pauser();
}
e = e - 1;
if (swapped == false) {
    break;
}
swapped = false;
for (var i = e - 1; i >= s; i--) {
    let rotate1 = bars[i].style.transform.match(/rotate\(((\d+)\d*)deg\)/)[1];
    let rotate2 = bars[i + 1].style.transform.match(/rotate\(((\d+)\d*)deg\)/)[1];
    let transY1 = bars[i].style.transform.match(/translateY\(((\d+)\d*)px\)/)[1];
    let transY2 = bars[i+1].style.transform.match(/translateY\(((\d+)\d*)px\)/)[1];
    var value1 = parseInt(bars[i].childNodes[0].innerHTML);
    var value2 = parseInt(bars[i + 1].childNodes[0].innerHTML);
    await new Promise((resolve) => {
        setTimeout(() => {
            resolve();
        }, timeoutValue)
    });
    bars[i].style.backgroundImage = "linear-gradient(90deg, pink, red)";
    bars[i + 1].style.backgroundImage = "linear-gradient(90deg, pink, red)";
    if (value1 > value2) {
        document.getElementById("messageDes").innerHTML = "If " + value1 + " > " + value2;
        var temp2 = bars[i].childNodes[0].innerText;
        await new Promise((resolve) => {
            setTimeout(() => {
                resolve();
            }, timeoutValue)
        });
        document.getElementById("messageDes").innerHTML = "Swap " + value1 + " and " +
value2;
        bars[i].style.transform = `translateX(${i * 40}px) translateY(${transY2}px)
rotate(${rotate2}deg)`;
        bars[i + 1].style.transform = `translateX(${(i + 1) * 40}px)
translateY(${transY1}px) rotate(${rotate1}deg)`;
        bars[i].childNodes[0].innerText = bars[i + 1].childNodes[0].innerText;
        bars[i + 1].childNodes[0].innerText = temp2;
        swapped = true;
    }else{
        document.getElementById("messageDes").innerHTML = "If " + value1 + " < " + value2
+ " => " + "do nothing" ;
    }
    await new Promise((resolve) => {
        setTimeout(() => {
            resolve();
        }, timeoutValue)
    });
}

```



```

bars[i].style.backgroundImage = "linear-gradient(90deg, #BFEFFF,#6FB3FF)";
    bars[i + 1].style.backgroundImage = "linear-gradient(90deg, #BFEFFF,#6FB3FF)";

    await waitforme(100);
    if (flag == 1) await pauser();
}
// Tăng giá trị s lên 1 vì phần tử đầu tiên đã được sắp xếp
s = s + 1;
}

for (var x = 0; x < 15; x++) {
    bars[x].style.backgroundImage = `linear-gradient(90deg, rgb(230, 240, 250), rgb(60,
120, 200))`;
}
document.getElementById("messageDes").innerHTML = "Sorted Complete!"
}
else if (number != 15) {
    let bars = document.querySelectorAll(".bar");
    // Khởi tạo biến swapped để kiểm tra việc hoán đổi
    let swapped = true;
    let sortTemp;
    // Khởi tạo biến s và e để giới hạn phạm vi duyệt
    let s = 0;
    let e = number;
    // Lặp lại cho đến khi không có hoán đổi nào được thực hiện
    while (swapped == true) {
        // Đặt lại giá trị swapped thành false
        swapped = false;
        slider.addEventListener("input", function () {
            timeoutValue = this.value;
        })
        // Duyệt từ trái sang phải và hoán đổi các phần tử không theo thứ tự
        for (var i = s; i < e - 1; i++) {
            var value1 = parseInt(bars[i].childNodes[0].innerHTML);
            var value2 = parseInt(bars[i + 1].childNodes[0].innerHTML);
            let rotate1 = bars[i].style.transform.match(/rotate\(((\[-]*\d+)\deg\))/[1];
            // Lấy giá trị rotate của phần tử thứ i+1
            let rotate2 = bars[i + 1].style.transform.match(/rotate\(((\[-]*\d+)\deg\))/[1];
            let transY1 = bars[i].style.transform.match(/translateY\(((\[-]\d+)\px\))/[1];
            let transY2 = bars[i+1].style.transform.match(/translateY\(((\[-]\d+)\px\))/[1];
            await new Promise((resolve) => {
                setTimeout(() => {
                    resolve();
                }, timeoutValue)
            });

```

```

bars[i].style.backgroundImage = "linear-gradient(90deg, pink, red)";
bars[i + 1].style.backgroundImage = "linear-gradient(90deg, pink,
red)";
    if (value1 > value2) {
        document.getElementById("messageDes").innerHTML = "If " + value1 +
" > " + value2;
        var temp2 = bars[i].childNodes[0].innerText;
        await new Promise((resolve) =>
            setTimeout(() => {
                resolve();
            }, timeoutValue)
        );
        c3.style.setProperty('--width', 0);
        document.getElementById("messageDes").innerHTML = "Swap " + value1
+ " and " + value2;
        bars[i].style.transform = `translateX(${i * 40}px)
translateY(${transY2}px) rotate(${rotate2}deg)`;
        bars[i + 1].style.transform = `translateX(${(i + 1) * 40}px)
translateY(${transY1}px) rotate(${rotate1}deg)`;
        bars[i].childNodes[0].innerText = bars[i +
1].childNodes[0].innerText;
        bars[i + 1].childNodes[0].innerText = temp2;
        // Đặt giá trị swapped thành true nếu có hoán đổi
        swapped = true;
    }else{
        document.getElementById("messageDes").innerHTML = "If " + value1 +
" < " + value2 + " => " + "do nothing" ;
    }
    await new Promise((resolve) =>
        setTimeout(() => {
            resolve();
        }, timeoutValue)
    );
};

```

```

bars[i].style.backgroundImage = "linear-gradient(90deg, #BFEFFF,#6FB3FF)";
bars[i + 1].style.backgroundImage = "linear-gradient(90deg,
#BFEFFF,#6FB3FF)";

```

```

        await waitforme(100);
        if (flag == 1) await pauser();
    }
    // Kiểm tra nếu không có hoán đổi nào trong lần duyệt này
    if (swapped == false) {
        break;
    }
}

```



```

document.getElementById("messageDes").innerHTML = "Swap " + value1 + " and " + value2;
    bars[i].style.transform = `translateX(${i * 40}px) translateY(${transY2}px)
rotate(${rotate2}deg)`;
    bars[i + 1].style.transform = `translateX(${(i + 1) * 40}px)
translateY(${transY1}px) rotate(${rotate1}deg)`;
    bars[i].childNodes[0].innerText = bars[i + 1].childNodes[0].innerText;
    bars[i + 1].childNodes[0].innerText = temp2;
    // Đặt lại giá trị swapped là true nếu có hoán đổi
    swapped = true;
} else {
    document.getElementById("messageDes").innerHTML = "If " + value1 + " < " +
value2 + " => " + "do nothing" ;
}
    await new Promise((resolve) => {
        setTimeout(() => {
            resolve();
        }, timeoutValue)
    });
    bars[i].style.backgroundImage = "linear-gradient(90deg, #BFEFFF,#6FB3FF)";
    bars[i + 1].style.backgroundImage = "linear-gradient(90deg, #BFEFFF,#6FB3FF)";

```

```

await waitforme(100);
    if (flag == 1) await pauser();
    }
    // Tăng biến s lên 1 vì phần tử đầu tiên đã được sắp xếp
    s = s + 1;
}
for (var x = 0; x < number; x++) {
    bars[x].style.backgroundImage = "linear-gradient(90deg, #16fb21, green)";
}

    document.getElementById("messageDes").innerHTML = "Sorted Complete!"
}
}

```

3.2.4. Hàm khởi tạo các ô nhập phân tử

```

let input_list = new Array();
function createInput() {
    let quantity = document.getElementById("quantity").value;
    document.getElementById("data_sort").innerHTML = "";
    for (let i = 1; i <= quantity; i++) {
        input_list[i] = document.createElement("INPUT");
        input_list[i].setAttribute("type", "number");
        input_list[i].setAttribute("style", "width: 56px");
        input_list[i].style.marginRight = "5px";
        input_list[i].style.marginBottom = "5px";
        input_list[i].style.marginTop = "5px";
        input_list[i].style.height = "25px";
        document.getElementById("data_sort").appendChild(input_list[i]);
    }
}

```

3.2.5. Hàm tạo dữ liệu từ bàn phím

```

function generateUserType() {
    const myElement = document.getElementById("container");
    myElement.innerHTML = '';
    let quantity = document.getElementById("quantity").value;
    for (let i = 1; i <= quantity; i++) {
        let value = input_list[i].value;
        let bar = document.createElement("div");
        bar.classList.add("bar");
        bar.style.height = `${value * 3}px`;
        bar.style.transform = `translateX(${i * 30}px)`;
        let barLabel = document.createElement("label");
        barLabel.classList.add("bar__id");
        barLabel.innerHTML = value;
        bar.appendChild(barLabel);
        container.appendChild(bar);
    }
}

```

3.2.6. Hàm quyết định tạo tự động hay từ bàn phím người dùng

```
function checkData() {
    let quantity = document.getElementById("quantity").value;
    for (let i = 1; i <= quantity; i++) {
        if (input_list[i].value == 0) {
            initialBars();
        } else {
            generateUserType();
        }
    }
    if (quantity == 0) {
        initialBars();
    }
}
```

3.2.7. Hàm xóa tập tin hiện tại

```
function checkFile() {
    if (document.getElementById("uploadFile").files.length != 0 ||
document.getElementById("uploadFile").files.length == 0) {
        $('#Button1-1').on('click', function (e) {
            var $element = $('#uploadFile');
            $element.wrap('<form>').closest('form').get(0).reset();
            $element.unwrap();
        });
        $('#Button6').on('click', function (e) {
            var $element = $('#uploadFile');
            $element.wrap('<form>').closest('form').get(0).reset();
            $element.unwrap();
        });
    }
}
```

3.2.8. Hàm tạm dừng hoặc tiếp tục thuật toán

```
document.getElementById("Button7").setAttribute("disabled", "true");
var flag = 0;

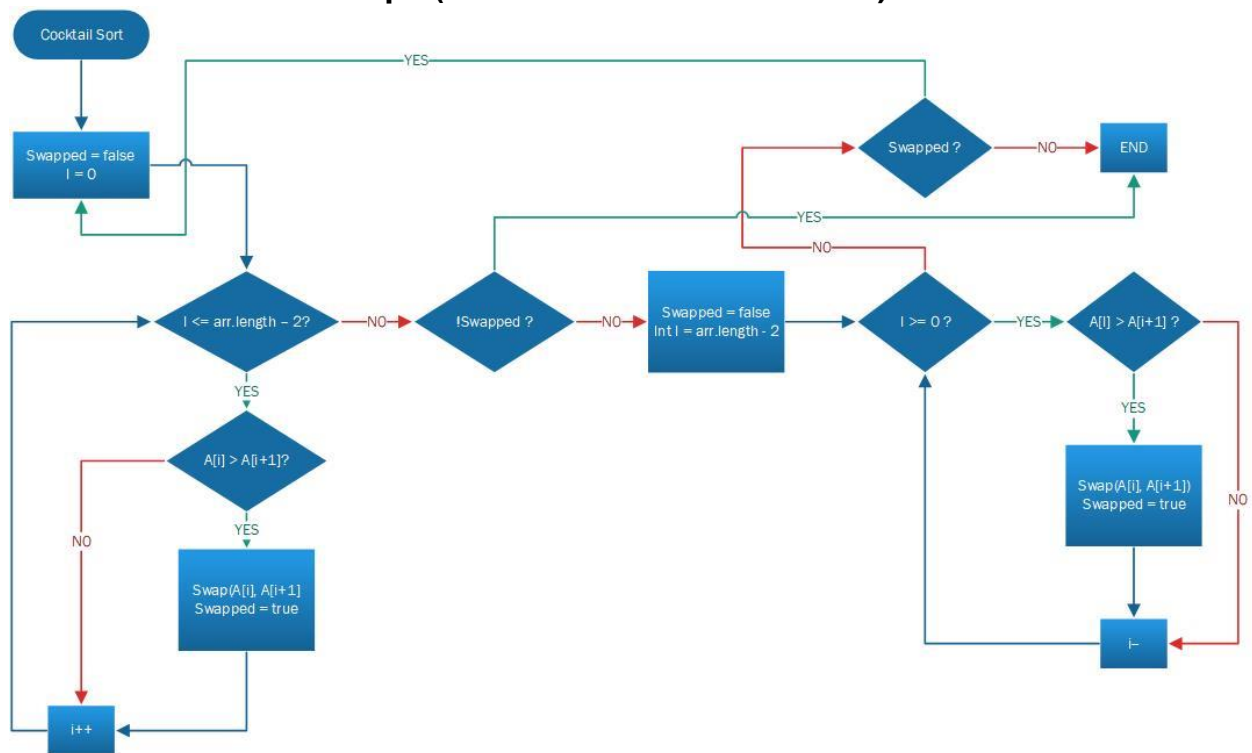
function waitforme(ms) {
    return new Promise(resolve => {
        setTimeout(() => {
            resolve('')
        }, ms);
    })
}

function pauser() {
    return new Promise(resolve => {
        let playbuttonclick = function () {
            document.getElementById("Button5").removeAttribute("disabled");
            document.getElementById("Button7").removeAttribute("disabled",
"true");
            document.getElementById("Button7").removeEventListener("click",
playbuttonclick);
            flag = 0;
            resolve("resolved");
        }
        document.getElementById("Button7").addEventListener("click",
playbuttonclick);
    })
}
```

3.2.9. Hàm ràng buộc ô “Số lượng”

```
function checkQuantity() {
    let number = document.getElementById('quantity').value;
    if (number < 0) {
        alert("Vui lòng nhập số lớn hơn 0");
    }
}
```

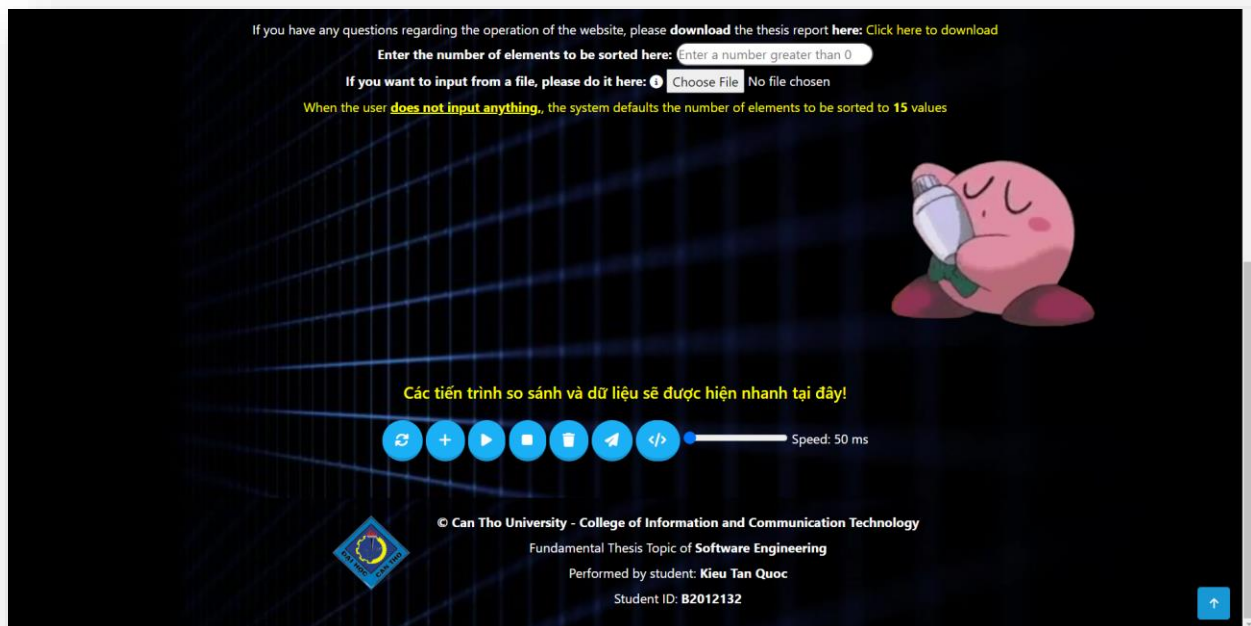
III. THIẾT KẾ GIẢI THUẬT (LƯỚI ĐỒ - NGÔN NGỮ GIẢ)



Lưu đồ giải thuật sắp xếp Cocktail








IV. GIỚI THIỆU TRANG WEB

3.4.1. Giao diện chính

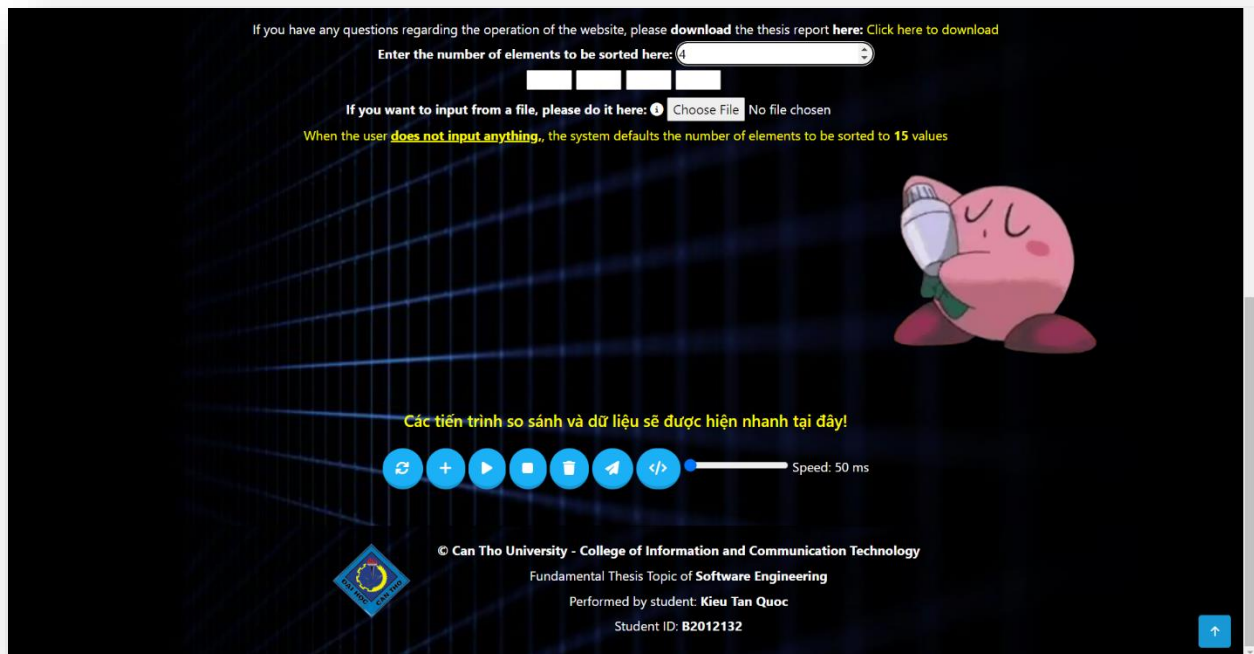


Giao diện chính gồm có 3 khu vực lớn: giới thiệu thuật toán và thực thi thuật toán, khu vực chức năng.

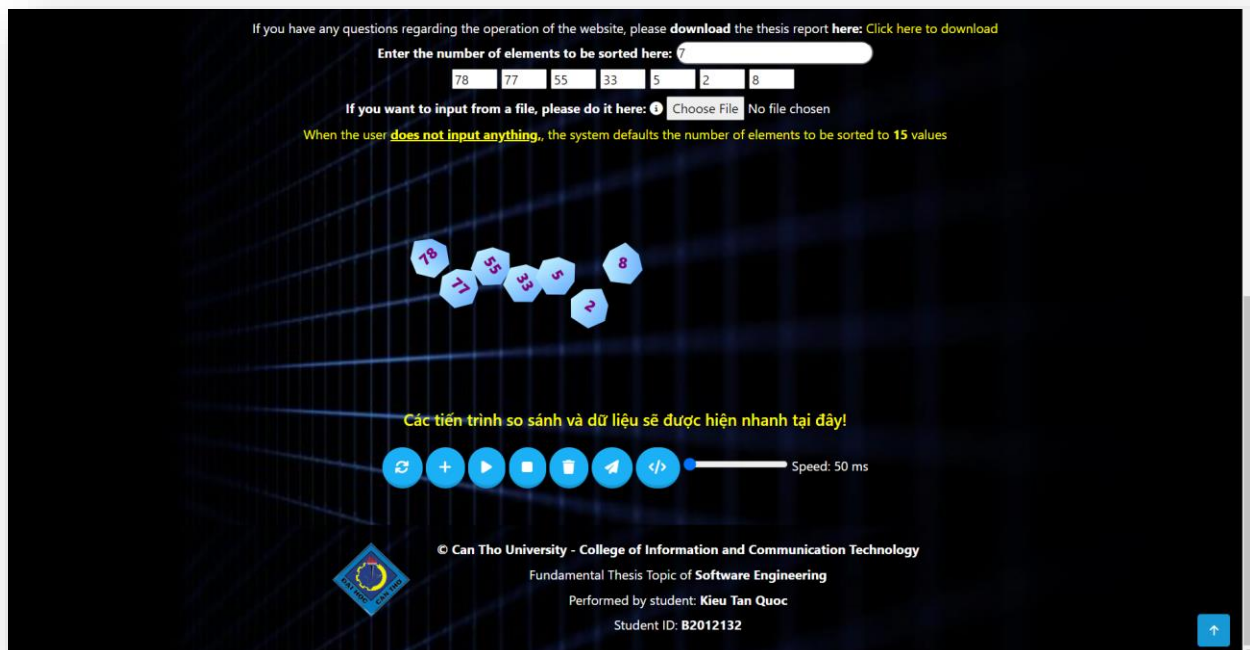
- Khu vực giới thiệu thuật toán: Bao gồm các thông tin cơ bản về thuật toán
- Khu vực thực thi thuật toán:
 - + Nút tải tài liệu: Tải tài liệu hướng dẫn sử dụng.
 - + Ô nhập: Dùng để nhập số lượng phần tử
 - + Nút chọn tập tin: Dùng cho phương pháp nhập dữ liệu từ tập tin
 - + Phần chứa dữ liệu chạy thuật toán: Dùng giao diện đồ họa để mô tả thuật toán.
- Khu vực chức năng:

	Dùng để <u>làm mới</u> lại một dữ liệu hiện có, không thể sử dụng khi không có dữ liệu.
	Dùng để tạo dữ liệu mới, khi dùng 1 trong 3 cách nhập dữ liệu. Khi người dùng không nhập số lượng phần tử, nút sẽ tạo ngẫu nhiên, khi người dùng nhập từ tập tin hoặc bàn phím, nút sẽ tạo theo dữ liệu đã cho.
	Dùng để chạy thuật toán, không thể thực hiện khi không có dữ liệu
	Có chức năng tạm dừng quá trình thực thi thuật toán.
	Có chức năng xóa dữ liệu đang chạy hiện tại.
	Có chức năng tiếp tục quá trình thực thi thuật toán.
	Hỗ trợ người dùng nhìn thấy quá trình thuật toán chạy và đoạn mã của thuật toán

3.4.2. Giao diện nhập dữ liệu từ bàn phím



3.4.3. Giao diện dữ liệu được hiển thị thông qua nhập từ bàn phím



CHƯƠNG 4. KẾT LUẬN – ĐÁNH GIÁ

I. KẾT QUẢ ĐẠT ĐƯỢC

4.1.1. Về mặt chuyên môn

Nắm vững kiến thức cấu trúc dữ liệu và giải thuật của thuật toán sắp xếp Cocktail. Hoàn tất được trang web “Mô phỏng thuật toán sắp xếp Cocktail” với các yêu cầu đã đề ra (có giao diện đồ họa, mô phỏng lại từng thao tác xử lý của thuật toán qua từng dòng code,...).

Học được kiến thức hàm xử lý bất đồng bộ, giúp việc thực thi trở nên nhanh hơn, tiết kiệm được nhiều thời gian.

Biết cách đưa website lên host để nhiều người có thể truy cập để sử dụng.

Biết cách sử dụng các thư viện Javascript (swiper, Vanilla,..) để tạo chuyển động cho trang web trở nên mượt mà hơn.

4.1.2. Về mặt kinh nghiệm

Biết cách sử dụng nhiều thư viện khác của Javascript, các loại hàm đặc biệt. Có thể phục vụ cho các đề tài nghiên cứu sau này.

Biết cách viết tài liệu theo mẫu chuẩn đã đề ra

Rèn luyện được khả năng quản lý thời gian, nhằm giúp cho đề tài niên luận kịp tiến độ.

Biết đánh giá giao diện dưới góc nhìn của người dùng thực tế.