

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO BÀI TẬP LỚN**  
**HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**Đề tài: Demonstration of Sorting Algorithm on  
Array (Bubble Sort, Quick Sort, Insertion Sort)**

Nhóm sinh viên thực hiện:

Trần Hoàng Dũng 20225708

Nguyễn Minh Đức 20225811

Lê Minh Đức 20225610

Vũ Thành Đạt 20225607

Phạm Minh Đức 20194517

Lớp : 151965

Giảng viên hướng dẫn: TS. Nguyễn Thị Thu Trang

*Hà Nội, tháng 12 năm 2024*

## LỜI CẢM ƠN

Lời đầu tiên, chúng em xin trân trọng cảm ơn và bày tỏ lòng biết ơn sâu sắc nhất tới cô **Nguyễn Thị Thu Trang** – Giảng viên Trường Công nghệ thông tin & Truyền thông, Trường Đại học Bách Khoa Hà Nội, giáo viên hướng dẫn bài tập lớn đã nhiệt tình giảng dạy, hướng dẫn, chỉ bảo.

Mặc dù đã cố gắng hoàn thiện sản phẩm nhưng không thể tránh khỏi những thiếu hụt về kiến thức. Chúng em mong muốn nhận được những nhận xét thẳng thắn, chi tiết đến từ thầy để tiếp tục hoàn thiện hơn nữa. Cuối cùng, chúng em xin được gửi lời cảm ơn đến cô Nguyễn Thị Thu Trang đã hướng dẫn chúng em trong suốt quá trình hoàn thiện bài tập lớn. Xin trân trọng cảm ơn cô.

***Xin chân thành cảm ơn!***

*Hà Nội, tháng 12 năm 2024*

# Table of Contents

<b>CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN.....</b>	<b>1</b>
<b>1.1 Mô tả yêu cầu bài toán.....</b>	<b>2</b>
<b>1.2 Yêu cầu cơ bản .....</b>	<b>2</b>
<b>Màn hình chính .....</b>	<b>2</b>
<b>Màn hình mô phỏng.....</b>	<b>2</b>
<b>CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ SƠ ĐỒ USE-CASE VÀ SƠ ĐỒ</b>	
<b>CLASS.....</b>	<b>3</b>
<b>2.1 Thiết kế sơ đồ class diagram.....</b>	<b>3</b>
<b>2.1.1 Utility.....</b>	<b>3</b>
<b>2.1.2 SortingAlgorithmController .....</b>	<b>4</b>
<b>2.1.3 QuickSortController.....</b>	<b>4</b>
<b>2.1.4 InsertionSortController.....</b>	<b>5</b>
<b>2.1.5 BubbleSortController .....</b>	<b>5</b>
<b>2.2 Thiết kế sơ đồ Use-case: .....</b>	<b>5</b>
<b>CHƯƠNG 3: DEMO VÀ ĐÁNH GIÁ KẾT QUẢ.....</b>	<b>7</b>
<b>3.1 DEMO .....</b>	<b>7</b>
<b>3.2 Đánh giá kết quả .....</b>	<b>8</b>
<b>PHÂN CHIA CÔNG VIỆC.....</b>	<b>9</b>

# CHƯƠNG 1. KHẢO SÁT, ĐẶC TẢ YÊU CẦU BÀI TOÁN

## 1.1 Mô tả yêu cầu bài toán

Mảng là cấu trúc cơ bản nhất trong khoa học máy tính. Hầu hết các phép toán cũng như các cấu trúc dữ liệu khác đều được xây dựng và thực hiện trên mảng. Trong dự án này, bạn sẽ phải tạo một ứng dụng để giải thích ba thuật toán sắp xếp: Sắp xếp nổi bọt (Bubble Sort), Sắp xếp nhanh (Quick Sort), Sắp xếp chèn (Insertion Sort).

## 1.2 Yêu cầu cơ bản

### Màn hình chính

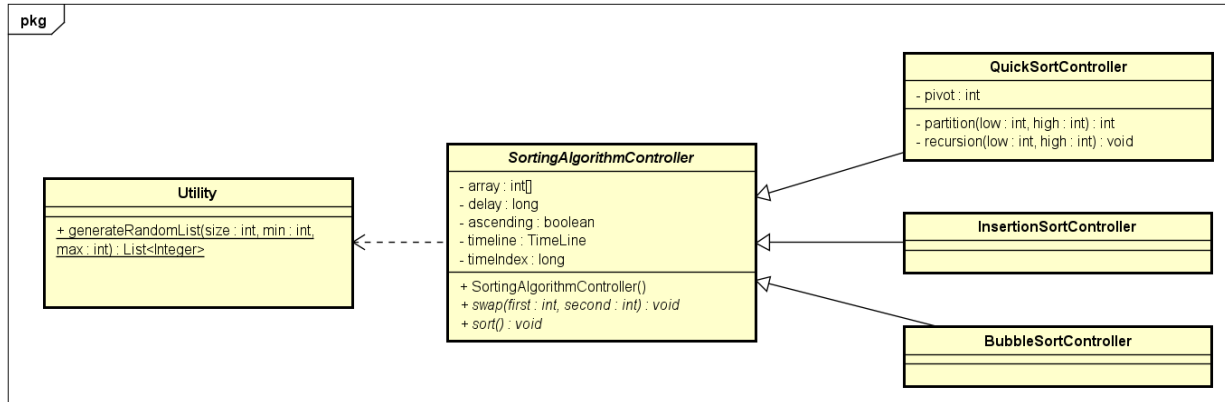
- Tiêu đề của chương trình
- Ba thuật toán sắp xếp cho người dùng chọn
- Nút HELP có chức năng hiển thị phần giới thiệu và hướng dẫn sử dụng chương trình
- Yêu cầu người dùng xác nhận thoát trước khi thoát khỏi chương trình

### Màn hình mô phỏng

- Một trường để tạo Array, người dùng có thể tự nhập mảng hoặc để hệ thống tạo ra một mảng ngẫu nhiên
- Một nút để bắt đầu thuật toán sắp xếp, cần hiện rõ từng bước sắp xếp
- Một nút để quay về màn hình chính

# CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ SƠ ĐỒ USE-CASE VÀ SƠ ĐỒ CLASS

## 2.1 Thiết kế sơ đồ class diagram



Tổng quan thiết kế:

- Utility: Lớp chứa phương thức tĩnh sử dụng để sinh mảng ngẫu nhiên
- SortingAlgorithmController: Lớp trừu tượng định nghĩa các thuộc tính và phương thức chung của từng lớp thực hiện sắp xếp mảng
- QuickSortController: Lớp con của lớp SortingAlgorithmController, có vai trò sắp xếp mảng sử dụng thuật toán sắp xếp nhanh (Quick sort), đồng thời cũng đóng vai trò làm controller cho giao diện của Quick sort.
- InsertionSortController: Lớp con của lớp SortingAlgorithmController, có vai trò sắp xếp mảng sử dụng thuật toán sắp xếp chèn (Insertion sort), đồng thời cũng đóng vai trò làm controller cho giao diện của Insertion sort.
- BubbleSortController: Lớp con của lớp SortingAlgorithmController, có vai trò sắp xếp mảng sử dụng thuật toán sắp xếp nổi bọt (Bubble sort), đồng thời cũng đóng vai trò làm controller cho giao diện của Bubble sort.

Mô tả chi tiết từng lớp:

### 2.1.1 Utility

Chức năng chính: Chứa các phương thức tĩnh dùng để hỗ trợ thực hiện một số chức năng

Phương thức:

- generateRandomList(size: int, min: int, max: int): List<Integer> - Thực hiện tạo ra một mảng ngẫu nhiên có độ dài là size và các phần tử nằm trong khoảng từ min đến max.

### 2.1.2 SortingAlgorithmController

Chức năng chính: Định nghĩa các thuộc tính và phương thức chung mà các lớp sắp xếp con sẽ tuân theo

Thuộc tính:

- array: int[] – mảng dùng để sắp xếp
- delay: long – Khoảng trễ giữa những lần thay đổi các phần tử trong mảng
- ascending: boolean – Quy định thứ tự sắp xếp (tăng dần hoặc giảm dần)
- timeIndex: long - Đếm số lần thực hiện thay đổi phần tử trong mảng
- timeline: TimeLine – Lưu trữ mỗi thay đổi của mảng

Phương thức:

- SortingAlgorithmController(): Phương thức khởi tạo
- swap(): void – Thực hiện trao đổi hai phần tử trong mảng
- sort(): void – Thực hiện sắp xếp

### 2.1.3 QuickSortController

Chức năng chính: Lớp con của SortingAlgorithmController. Thực hiện sắp xếp mảng sử dụng thuật toán sắp xếp nhanh.

Thuộc tính:

- pivot: int – Lưu phần tử chốt của phân đoạn đang thực hiện
- partition(low: int, high: int): int – Hàm sẽ chọn một phần tử chốt (pivot), mục tiêu là di chuyển các phần tử lớn hơn pivot sang phải pivot, và các phần tử nhỏ hơn pivot sang trái pivot nếu sắp xếp theo thứ tự tăng dần và ngược lại.
- recursion(low: int, high: int): void – Hàm đệ quy được sử dụng để thực hiện gọi đệ quy theo phương pháp chia để trị.

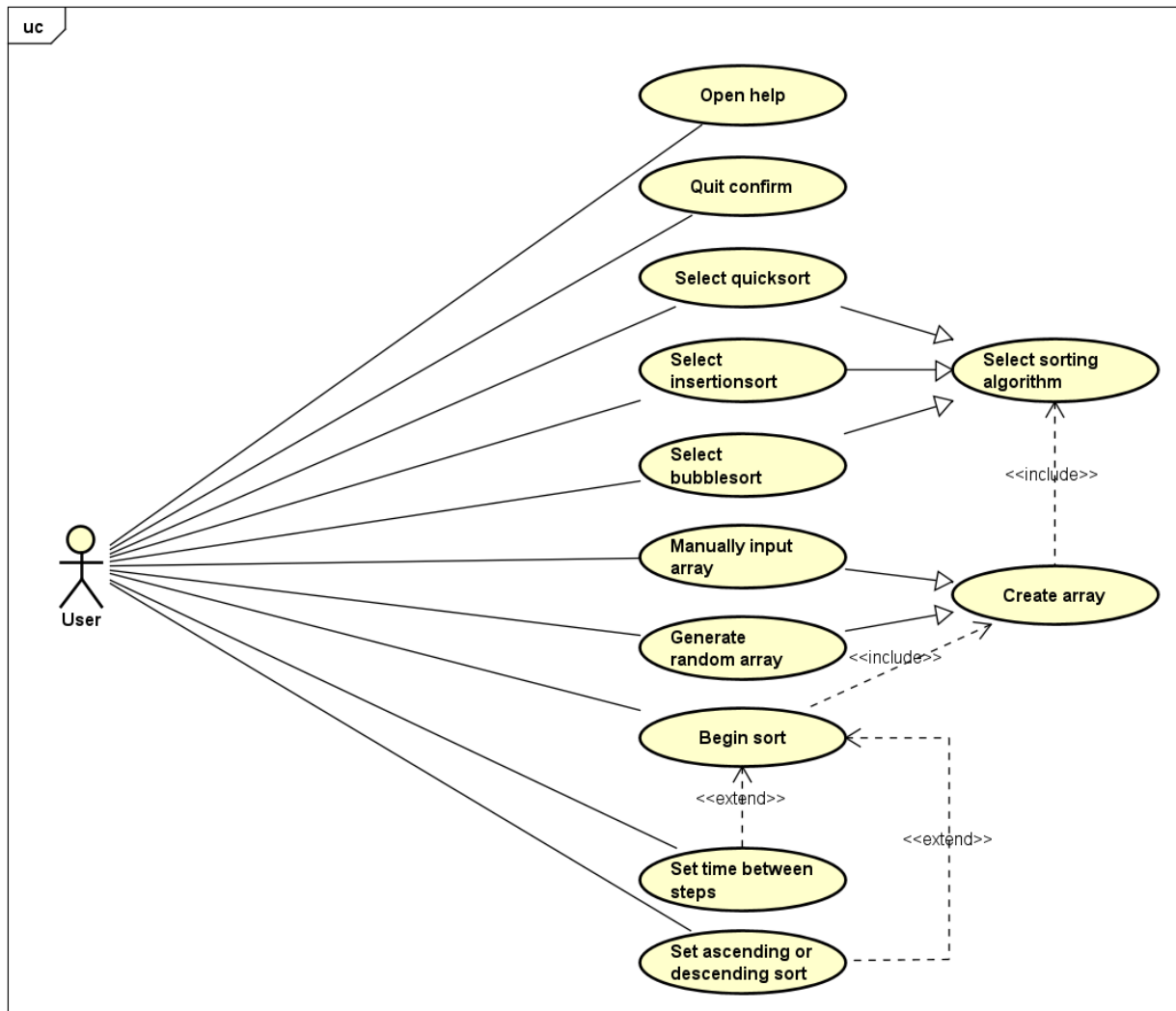
### 2.1.4 InsertionSortController

Chức năng chính: Lớp con của SortingAlgorithmController. Thực hiện sắp xếp mảng sử dụng thuật toán sắp xếp chèn.

### 2.1.5 BubbleSortController

Chức năng chính: Lớp con của SortingAlgorithmController. Thực hiện sắp xếp mảng sử dụng thuật toán sắp xếp nổi bọt.

## 2.2 Thiết kế sơ đồ Use-case:



Đặc tả các Use-case:

- Open help: Người dùng mở phần hỗ trợ

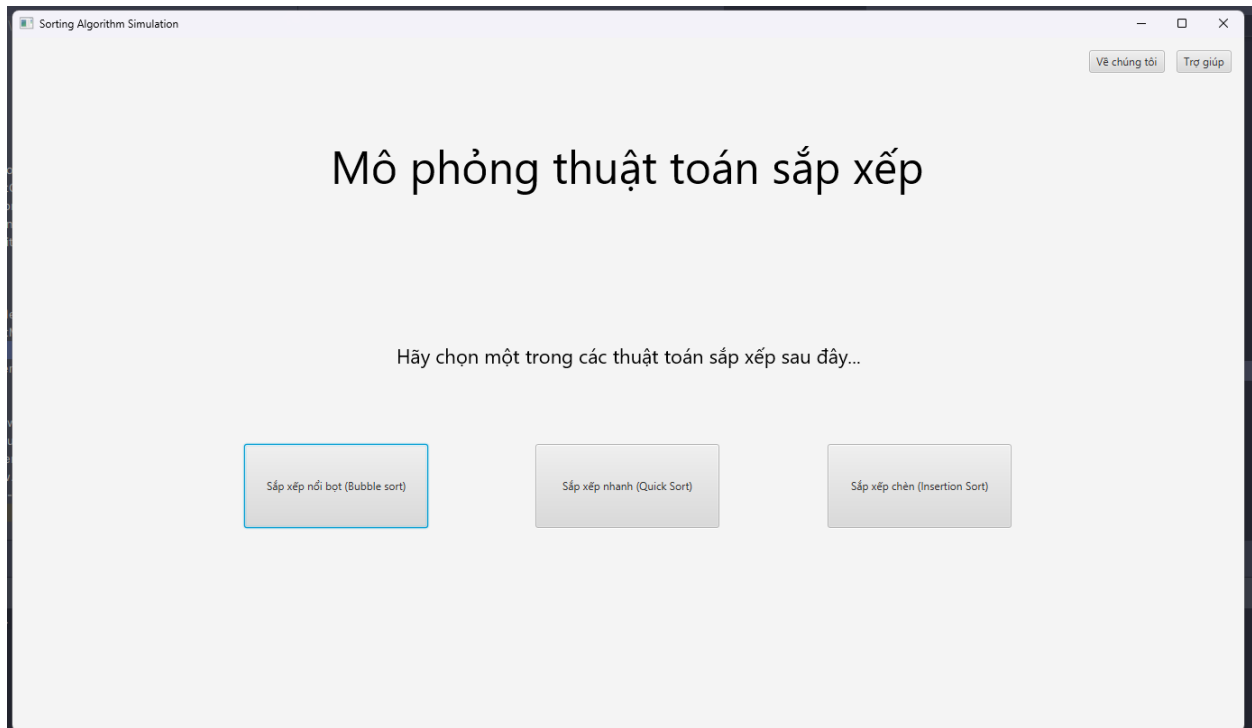
- Quit confirm: Người dùng phải xác nhận thoát trước khi thoát khỏi chương trình
- Select sorting algorithm: Người dùng chọn thuật toán sắp xếp bằng cách chọn một trong ba thuật toán: Select quick sort, select insertion sort, select bubble sort
- Create array: Yêu cầu người dùng trước đó phải chọn thuật toán sắp xếp. Lúc này người dùng tạo mảng, use-case này có thể được thực hiện bởi người dùng theo hai cách, hoặc tự nhập mảng vào từ bàn phím (Manually input array) hoặc tạo mảng ngẫu nhiên (Generate random array).
- Begin sort: Yêu cầu người dùng phải tạo mảng. Lúc này người dùng có thể thực hiện sắp xếp trên mảng đã được tạo.
- Set time between steps: Người dùng đặt thời gian nghỉ giữa mỗi bước thay đổi mảng, mặc định là 500 ms.
- Set ascending or descending sort: Người dùng chọn thứ tự sắp xếp, tăng dần hoặc giảm dần.



## CHƯƠNG 3: DEMO VÀ ĐÁNH GIÁ KẾT QUẢ

### 3.1 DEMO

Khi khởi động chương trình người dùng sẽ thấy cửa sổ gồm 3 nút tương ứng với 3 thuật toán sắp xếp ở giữa, và 2 nút ở góc trên bên phải hiển thị thông tin về đội ngũ tạo ra chương trình và phần hỗ trợ.



Khi nhấn vào một trong các thuật toán sắp xếp người dùng sẽ được chuyển đến giao diện của thuật toán tương ứng. Lúc này người dùng có thể thực hiện tạo mảng, sau đó thực hiện sắp xếp mảng. Người dùng còn có thể chọn sắp xếp tăng dần hoặc giảm dần thông qua việc đánh vào ô thứ tự, và điều chỉnh thời gian chờ giữa các lần thay đổi nhằm giúp cho việc quan sát sự thay đổi dễ dàng hơn.

### 3.2 Đánh giá kết quả

Chương trình đã mô phỏng được ba thuật toán sắp xếp (sắp xếp nhanh, sắp xếp nổi bọt, sắp xếp chèn) với một số tính năng hỗ trợ giúp cho việc theo dõi sự thay đổi của mảng một cách dễ dàng. Tuy nhiên vẫn còn một số hạn chế do các thành viên trong nhóm chưa có nhiều kinh nghiệm trong việc phát triển ứng dụng bằng Java cũng như thiết kế giao diện người dùng.

Nhóm chúng em xin tóm tắt lại một số ưu nhược điểm như sau:

Ưu điểm:

- Chương trình đáp ứng yêu cầu cơ bản của bài toán
- Chương trình có thêm một số tính năng nhằm cải thiện trải nghiệm người dùng

Nhược điểm:

- Giao diện chưa bắt mắt
- Đối với các mảng có kích thước lớn thì thiết kế hiện tại có thể khiến người dùng khó theo dõi được sự thay đổi của mảng

# PHÂN CHIA CÔNG VIỆC

Trần Hoàng Dũng 20225708 (TL)

- Class design: 100%
- Quicksort: 100%
- Report: 33.34%
- Release Flow: 50%

Nguyễn Minh Đức 20225811

- Use-case design: 100%
- Insertionsort: 100%
- UI: 25%
- Release Flow: 50%

Lê Minh Đức 20225610

- Testing: 33.33%
- Bubblesort: 100%
- UI: 25%

Vũ Thành Đạt 20225607

- UI: 25%
- Report: 33.33%
- Testing: 33.33%

Phạm Minh Đức 20194517

- Report: 33.33%
- Testing: 33.34%
- UI: 25%