**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC PHÚ XUÂN**

****

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN HỌC PHẦN**

**Cấu Trúc Dữ Liệu Và Giải Thuật**

**a**

**Giảng viên hướng dẫn: Châu Thị Dung**

**Sinh viên thực hiện: Lê Tấn Quốc Đạt**

**Lớp: CNTT K21**

**Học Kỳ: Fall**

**Năm học: 2024-2025**

**HUẾ, 11/2024**

# Mục Lục

[Mục Lục 1](#_Toc182349926)

[LỜI GIỚI THIỆU 1](#_Toc182349927)

[CHƯƠNG 1 :TỔNG QUAN ĐỀ TÀI 2](#_Toc182349928)

[1. Giới thiệu đề tài 2](#_Toc182349929)

[2. Mục tiêu nghiên cứu 2](#_Toc182349930)

[3. Phạm vi thực hiện 2](#_Toc182349931)

[4. Tính năng của hệ thống 3](#_Toc182349932)

[5. Công cụ sử dụng : 4](#_Toc182349933)

[CHƯƠNG 2 :CHƯƠNG TRÌNH 5](#_Toc182349934)

[2.1. Các chức măng thực hiện : 5](#_Toc182349935)

[2.2. Giao diện chương trình : 7](#_Toc182349936)

[CHƯƠNG 3 :KẾT LUẬN 15](#_Toc182349942)

[3.1. Kết quả đạt được: 15](#_Toc182349943)

[3.2. Hướng phát triển đồ án : 17](#_Toc182349944)

[TÀI LIỆU THAM KHẢO 19](#_Toc182349945)

# LỜI GIỚI THIỆU

Em rất hân hạnh được học môn Cấu Trúc Dữ Liệu và Thuật Toán dưới sự hướng dẫn tận tình của cô Châu Thị Dung. Môn học này đã mở ra cho em một thế giới mới về cách tổ chức và xử lý dữ liệu trong lĩnh vực khoa học máy tính. Cô đã giúp em hiểu rõ tầm quan trọng của việc nắm vững các cấu trúc dữ liệu cơ bản như mảng, danh sách liên kết, ngăn xếp và hàng đợi, cũng như các cấu trúc phức tạp hơn như cây và đồ thị.

Trong suốt khóa học, cô Dung không chỉ truyền đạt kiến thức lý thuyết mà còn tạo nhiều cơ hội cho em thực hành. Em đã được áp dụng các thuật toán đệ quy để giải quyết các bài toán như tính giai thừa, dãy Fibonacci và Tháp Hà Nội. Cô cũng hướng dẫn em tìm hiểu về các thuật toán sắp xếp quan trọng như sắp xếp nổi bọt, sắp xếp chọn và sắp xếp nhanh, giúp em hiểu sâu hơn về cách tối ưu hóa việc xử lý dữ liệu, cô giao cho em nhiệm vụ viết chương trình quản lý sinh viên sử dụng cấu trúc dữ liệu danh sách liên kết và cây. Thông qua dự án này, em không chỉ củng cố kiến thức đã học mà còn phát triển kỹ năng lập trình và tư duy logic của mình. Cô luôn khuyến khích em sáng tạo và áp dụng kiến thức vào thực tế, điều này đã giúp em tự tin hơn trong việc giải quyết các vấn đề phức tạp. Em tin rằng những gì học được từ cô Dung sẽ là nền tảng vững chắc cho sự nghiệp tương lai của em trong ngành công nghệ thông tin. Em xin chân thành cảm ơn cô đã truyền cảm hứng và định hướng cho em trong suốt khóa học này.

# CHƯƠNG 1 :TỔNG QUAN ĐỀ TÀI

## ****1. Giới thiệu đề tài****

Trong lĩnh vực khoa học máy tính, việc áp dụng các thuật toán và cấu trúc dữ liệu để giải quyết các bài toán thực tiễn là vô cùng quan trọng. Đề tài này phát triển một chương trình tích hợp các tính năng từ giải thuật đệ quy, quản lý sinh viên bằng cấu trúc cây nhị phân, đến tìm đường đi ngắn nhất trong đồ thị. Với các tính năng trên, chương trình không chỉ phục vụ mục đích học tập mà còn có khả năng mở rộng để ứng dụng trong quản lý dữ liệu và tối ưu hóa.

Mục tiêu của đề tài là xây dựng một hệ thống đơn giản và hiệu quả, giúp người dùng thao tác dễ dàng thông qua các menu lựa chọn. Hệ thống không chỉ đáp ứng các yêu cầu cơ bản mà còn cung cấp một công cụ trực quan để làm quen với cấu trúc dữ liệu, các thuật toán đệ quy và một số thuật toán nâng cao.

## ****2. Mục tiêu nghiên cứu****

* Tìm hiểu và áp dụng các bài toán đệ quy cơ bản.
* Quản lý dữ liệu sinh viên sử dụng cấu trúc cây nhị phân.
* Áp dụng thuật toán Dijkstra để tìm đường đi ngắn nhất.
* Thiết kế giao diện menu đơn giản, thân thiện với người dùng.

## ****3. Phạm vi thực hiện****

Chương trình sẽ bao gồm các chức năng chính sau:

1. **Bài toán đệ quy**: Giải quyết các bài toán đệ quy cơ bản như tính giai thừa, dãy Fibonacci, tìm phần tử lớn nhất trong mảng, chuyển đổi hệ cơ số, và các thuật toán sắp xếp.
2. **Quản lý sinh viên bằng cấu trúc cây nhị phân**: Thực hiện các thao tác cơ bản với dữ liệu sinh viên như thêm, xóa, sửa, tìm kiếm, và thống kê số lượng sinh viên đạt yêu cầu. Hệ thống sử dụng cây nhị phân để quản lý dữ liệu một cách có tổ chức và hiệu quả.
3. **Tìm đường đi ngắn nhất**: Áp dụng thuật toán Dijkstra để tìm đường đi ngắn nhất trong một đồ thị, phục vụ cho các bài toán tối ưu hóa.
4. **Giao diện menu chính và các menu phụ**: Cho phép người dùng dễ dàng lựa chọn và thực hiện các chức năng của chương trình.

## ****4. Tính năng của hệ thống****

### 4.1 Menu chính

* **Bài toán đệ quy**: Cho phép người dùng truy cập vào danh sách các bài toán đệ quy, cung cấp các phương pháp giải quyết từ cơ bản đến nâng cao.
* **Danh sách liên kết đôi**: Quản lý danh sách sinh viên thông qua danh sách liên kết đôi, cho phép cập nhật và truy xuất dữ liệu linh hoạt.
* **Cây nhị phân quản lý sinh viên**: Các thao tác cơ bản như thêm, hiển thị, cập nhật và xóa sinh viên, cùng với các tính năng thống kê, lọc và hiển thị sinh viên có điểm >= 5.
* **Tìm đường đi ngắn nhất**: Cung cấp công cụ tìm đường đi ngắn nhất trong một đồ thị bằng thuật toán Dijkstra.

### 4.2 Menu bài toán đệ quy

* **Các bài toán cơ bản**:
  + **Giai thừa**: Tính giai thừa của một số nguyên dương bằng phương pháp đệ quy.
  + **Dãy Fibonacci**: Tính số Fibonacci thứ n bằng phương pháp đệ quy.
  + **Tìm phần tử lớn nhất trong mảng**: Sử dụng đệ quy để tìm phần tử lớn nhất.
  + **Chuyển đổi thập phân sang nhị phân**: Chuyển đổi một số thập phân sang nhị phân.
  + **Đếm số chữ số của một số**: Đếm số lượng chữ số của một số nguyên.
  + **Tìm tất cả các tổ hợp của mảng**: Tạo ra các tổ hợp có thể có của các phần tử trong mảng.
* **Bài toán nâng cao**:
  + **N-Queen**: Giải bài toán N quân hậu bằng đệ quy.
  + **Sudoku**: Giải bài toán Sudoku bằng đệ quy.
* **Thuật toán sắp xếp**:
  + **Sắp xếp nổi bọt (Bubble Sort)**
  + **Sắp xếp chèn (Insertion Sort)**
  + **Sắp xếp trộn (Merge Sort)**
  + **Sắp xếp nhanh (Quick Sort)**

### 4.3 Menu cây nhị phân quản lý sinh viên

* **Thêm sinh viên**: Thêm thông tin của sinh viên vào hệ thống.
* **Hiển thị sinh viên**: Xem danh sách sinh viên hiện có trong hệ thống.
* **Cập nhật sinh viên**: Chỉnh sửa thông tin của sinh viên.
* **Xóa sinh viên**: Xóa thông tin của sinh viên khỏi hệ thống.
* **Thống kê đậu/rớt**: Kiểm tra kết quả đậu hoặc rớt của sinh viên.
* **Hiển thị sinh viên có điểm >= 5**: Lọc và hiển thị sinh viên đạt điểm từ 5 trở lên.

## Công cụ sử dụng :

**Để phát triển hệ thống, các công cụ và công nghệ sau đây được sử dụng:**

* **Ngôn ngữ lập trình Java**: Dùng để xây dựng các chức năng chính của hệ thống, xử lý đệ quy, khử đệ quy, và các cấu trúc dữ liệu như danh sách liên kết, cây nhị phân và đồ thị.
* **Eclipse/IntelliJ IDEA**: Môi trường phát triển tích hợp (IDE) để viết và quản lý mã nguồn.
* **Java Collections Framework**: Sử dụng để làm việc với các cấu trúc dữ liệu như danh sách, cây, và bảng băm.
* **File I/O trong Java**: Đọc và ghi dữ liệu vào file để lưu trữ thông tin sinh viên và các kết quả tính toán.
* **JUnit**: Để kiểm thử các hàm và đảm bảo hệ thống hoạt động đúng.
* **Git/GitHub**: Để quản lý phiên bản mã nguồn và làm việc nhóm.
* **Tài liệu PDF/Word**: Dùng để tạo tài liệu hướng dẫn sử dụng và báo cáo kết quả.

# CHƯƠNG 2 :CHƯƠNG TRÌNH

## 2.1. Các chức măng thực hiện :

**Hệ thống được thiết kế với các chức năng chính như sau:**

**Giải quyết các bài toán Đệ Quy và Khử Đệ Quy:**

1. **Đệ quy:**

* Các bài toán cơ bản:
  + 1. Giai thừa: Tính giai thừa của một số nguyên dương bằng phương pháp đệ quy.
    2. Dãy Fibonacci: Tính số Fibonacci thứ n bằng phương pháp đệ quy.
    3. Tìm phần tử lớn nhất trong mảng: Sử dụng đệ quy để tìm phần tử lớn nhất.
    4. Chuyển đổi thập phân sang nhị phân: Chuyển đổi một số thập phân sang nhị phân.
    5. Đếm số chữ số của một số: Đếm số lượng chữ số của một số nguyên.
    6. Tìm tất cả các tổ hợp của mảng: Tạo ra các tổ hợp có thể có của các phần tử trong mảng.
* Bài toán nâng cao: ( Đệ quy quay lùi )
  + 1. N-Queen: Giải bài toán N quân hậu bằng đệ quy.
    2. Sudoku: Giải bài toán Sudoku bằng đệ quy.
* Thuật toán sắp xếp:
  + 1. Sắp xếp nổi bọt (Bubble Sort)
    2. Sắp xếp chèn (Insertion Sort)
    3. Sắp xếp trộn (Merge Sort)
    4. Sắp xếp nhanh (Quick Sort)

1. **Khử đệ quy:**
   * 1. Các bài toán trên được chuyển đổi sang phương pháp không đệ quy (lặp) để minh họa cách xử lý bài toán đệ quy bằng phương pháp khác.

**Quản lý sinh viên bằng Danh sách liên kết đôi:**

* **Danh sách liên kết đôi:**

1. Thêm sinh viên: Thêm thông tin của sinh viên vào hệ thống.
2. Hiển thị sinh viên: Xem danh sách sinh viên hiện có trong hệ thống.
3. Cập nhật sinh viên: Chỉnh sửa thông tin của sinh viên.
4. Xóa sinh viên: Xóa thông tin của sinh viên khỏi hệ thống.
5. Thống kê đậu/rớt: Kiểm tra kết quả đậu hoặc rớt của sinh viên.
6. Hiển thị sinh viên có điểm >= 5: Lọc và hiển thị sinh viên đạt điểm từ 5 trở lên.

**Quản lý sinh viên bằng Cây nhị phân (Binary Tree):**

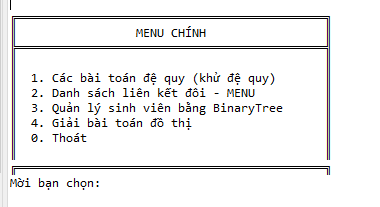
1. **Thêm sinh viên: Nhập mã, tên, điểm để thêm vào cây.**
2. **Hiển thị sinh viên: Xem danh sách toàn bộ sinh viên với điểm, xếp loại, kết quả.**
3. **Cập nhật sinh viên: Sửa thông tin sinh viên theo mã.**
4. **Xóa sinh viên: Xóa sinh viên theo mã.**
5. **Thống kê đậu/rớt: Xem số sinh viên đậu/rớt và chi tiết.**
6. **Hiển thị sinh viên có điểm >= 5: Xem danh sách sinh viên đạt điểm từ 5 trở lên.**
7. **Thoát và lưu: Lưu tự động dữ liệu khi thoát về menu chính.**

**Giao diện Menu dễ sử dụng:**

* 1. Giao diện dòng lệnh thân thiện giúp người dùng dễ dàng điều hướng qua các chức năng khác nhau.
  2. Các **menu phụ** tương ứng với từng chức năng chính, giúp người dùng thực hiện các thao tác một cách hiệu quả.

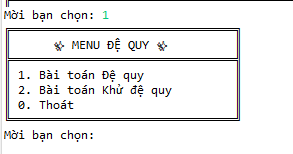
## 2.2. Giao diện chương trình :

#### ****Menu Chính:****



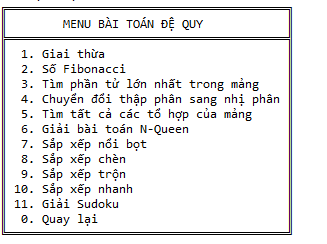
#### ****Menu Đệ Quy và Khử Đệ Quy:****

* Khi người dùng chọn chức năng liên quan đến các bài toán đệ quy, giao diện sẽ chuyển đến **Menu Đệ Quy Và Khử Đệ Quy**.

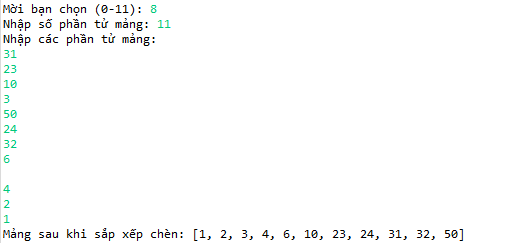


#### ****Menu Các Bài Đệ Quy:****

* Tại đây, người dùng có thể chọn các bài toán Đệ Quy (Khử Đệ Quy) ( hoặc có thể đệ quy quay lui) cụ thể như tính giai thừa, dãy Fibonacci, hoặc các thuật toán sắp xếp ,..

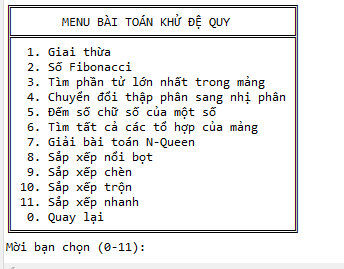


Ví dụ: khì ta sắp xếp chèn sẽ được như vậy



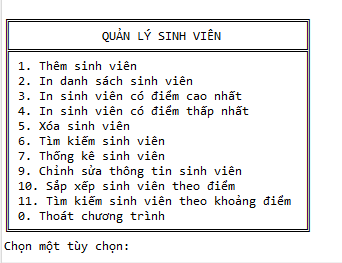
Khử đệ quy cũng tương tự như vậy , các bài toán đã chuyển đổi sang phương pháp không đệ quy:

#### ****Menu Khử Đệ Quy:****

* Nếu chọn phần "Khử đệ quy", Khử đệ quy cũng tương tự như vậy , các bài toán đã chuyển đổi sang phương pháp không đệ quy 

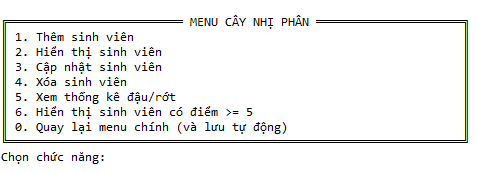
**Giao Diện Quản Lý Danh Sách Liên Kết Đôi:**

* Giao diện hiển thị danh sách các tùy chọn và yêu cầu nhập từ người dùng để thực hiện các thao tác cần thiết.
* Các chức năng trên giúp xây dựng một hệ thống quản lý sinh viên toàn diện và tiện lợi, giúp người dùng dễ dàng quản lý và tìm kiếm thông tin sinh viên, đồng thời hỗ trợ trong việc thống kê và phân loại sinh viên một cách hiệu quả.

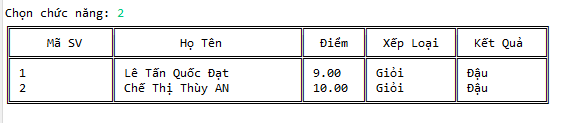


**Giao Diện Quản Lý Sinh Viên Bằng Cây Nhị Phân (Binary Tree):**

* Người dùng sẽ thấy các tùy chọn để nhập sinh viên, duyệt cây, xóa, tìm kiếm hoặc xuất dữ liệu ra file.
* Giao diện menu quản lý cây nhị phân cung cấp các tùy chọn để người dùng dễ dàng thao tác và quản lý thông tin sinh viên.



Xem những thống kê :



**Giao Diện Giải Bài Toán Đồ Thị:**

# Cấu trúc dữ liệu và giao diện cho ứng dụng tìm đường đi ngắn nhất

## Cấu trúc dữ liệu

* Sử dụng đồ thị để biểu diễn mạng lưới các điểm và đường đi
* Lưu trữ đỉnh và cạnh của đồ thị
* Cài đặt thuật toán tìm đường đi ngắn nhất (như Dijkstra hoặc A\*)

## Giao diện người dùng

* Giao diện cho phép người dùng tạo đồ thị, thêm đỉnh và cạnh, tính toán và hiển thị đường đi ngắn nhất giữa các điểm trong đồ thị.
* Các tùy chọn được thiết kế giúp người dùng dễ dàng nhập liệu và xem kết quả đường đi ngắn nhất.

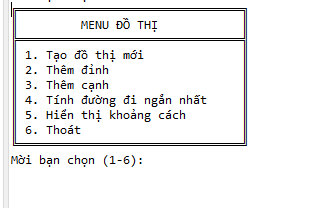
## Các chức năng chính

1. Tạo và chỉnh sửa đồ thị
2. Thêm và xóa đỉnh
3. Thêm và xóa cạnh
4. Tính toán đường đi ngắn nhất
5. Hiển thị kết quả đường đi

## Cải tiến tiềm năng

* Thêm tính năng lưu và tải đồ thị
* Hỗ trợ nhiều thuật toán tìm đường khác nhau
* Tích hợp với b

Ví Dụ:



**Tạo đồ thị mới:**

* Nhập số đỉnh của đồ thị: 5

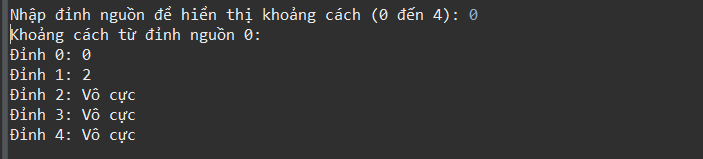
**Thêm cạnh:**

* Nhập đỉnh bắt đầu (0 đến 4): 0
* Nhập đỉnh kết thúc (0 đến 4): 1
* Nhập trọng số: 2

**Tính đường đi ngắn nhất:**

* Nhập đỉnh nguồn cho Dijkstra (0 đến 4): 0

**Hiển thị kết quả:**



# 

# CHƯƠNG 3 :KẾT LUẬN

## 3.1. Kết quả đạt được:

* Xây dựng thành công cấu trúc dữ liệu đồ thị để biểu diễn mạng lưới điểm và đường đi
* Triển khai hiệu quả thuật toán tìm đường đi ngắn nhất (Dijkstra hoặc A\*)
* Phát triển giao diện người dùng tương tác cho phép thêm/chỉnh sửa đồ thị và hiển thị kết quả
* Tích hợp các chức năng chính như tạo/chỉnh sửa đồ thị, tính toán và hiển thị đường đi ngắn nhất
* Thêm tính năng so sánh nhiều đường đi và đặt trọng số cho các cạnh
* Cải thiện hiệu suất và độ chính xác của thuật toán tìm đường
* **Hoàn thiện chức năng chính**: Đồ án đã hoàn thành các chức năng chính như quản lý đệ quy, danh sách liên kết đôi, quản lý sinh viên bằng cây nhị phân, và bài toán đồ thị về tìm đường đi ngắn nhất

Người dùng có thể thực hiện các thao tác cơ bản như thêm, xóa, tìm kiếm, và hiển thị thông tin.

* **Giao diện người dùng**: Giao diện menu rõ ràng và dễ sử dụng, giúp người dùng nhanh chóng lựa chọn chức năng mà họ cần mà không gặp khó khăn.
* **Xử lý lỗi**: Đã triển khai một số kiểm tra đầu vào cơ bản để đảm bảo rằng người dùng nhập dữ liệu hợp lệ, giảm thiểu tình trạng lỗi trong quá trình chạy chương trình.
* **Tính hiệu quả**: Các thuật toán đã được tối ưu hóa để đảm bảo hiệu suất tốt, đặc biệt là trong các thao tác tìm kiếm và sắp xếp.
* **Tài liệu hướng dẫn**: Cung cấp tài liệu hướng dẫn sử dụng và mã nguồn giúp nguời dùng dễ dàng hiểu và áp dụng.

## 3.2. Hướng phát triển đồ án :

* **1. Tối ưu hóa thuật toán**
  + **Cải thiện hiệu suất của thuật toán tìm đường đi ngắn nhất**
  + **Nghiên cứu và áp dụng các thuật toán mới, hiệu quả hơn**
* **2. Mở rộng chức năng**
  + **Thêm tính năng tìm đường đi có điểm dừng trung gian**
  + **Phát triển chức năng tìm đường đi tối ưu dựa trên nhiều tiêu chí (thời gian, chi phí, khoảng cách)**
* **3. Cải thiện giao diện người dùng**
  + **Thiết kế giao diện đồ họa trực quan và thân thiện hơn**
  + **Thêm tính năng tương tác và phản hồi thời gian thực**
* **4. Tích hợp dữ liệu thực tế**
  + **Kết nối với dữ liệu giao thông thời gian thực**
  + **Tích hợp bản đồ thực tế và hệ thống định vị GPS**
* **5. Mở rộng phạm vi ứng dụng**
  + **Phát triển phiên bản di động của ứng dụng**
  + **Tạo API để tích hợp với các hệ thống khác**
* **6. Nâng cao độ chính xác**
  + **Cải thiện độ chính xác của mô hình đồ thị**
  + **Thêm các yếu tố thực tế như tắc nghẽn giao thông, công trình xây dựng**
* **7. Bảo mật và quản lý dữ liệu**
  + **Tăng cường bảo mật cho dữ liệu người dùng**
  + **Phát triển hệ thống sao lưu và khôi phục dữ liệu**

# 

# TÀI LIỆU THAM KHẢO

**1.https://www.youtube.com/**

**2.https://chatgpt.com/**

**3.https://www.vietjack.com/cau-truc-du-lieu-va-giai-thuat/index.jsp**