**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

** KHOA CÔNG NGHỆ THÔNG TIN**

**THỰC TẬP CƠ SỞ CÔNG NGHỆ THÔNG TIN**

**Đề tài: Thao tác trên cây nhị phân tìm kiếm**

**Giảng viên hướng dẫn:Nguyễn Thị Hương Lý**

**Sinh viên thực hiện: Trịnh Minh Hậu**

**Mã số sinh viên: 61133622**

**Khánh Hòa – 2022**

Mục Lục

[1. Lý do chọn đề tài: 3](#_Toc92385175)

[2. Mục tiêu của đề tài. 3](#_Toc92385176)

[3. Phạm vi nghiên cứu. 3](#_Toc92385177)

[4. Phương pháp nghiên cứu. 3](#_Toc92385178)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 4](#_Toc92385179)

[1.Lý thuyết về cây tìm kiếm nhị phân: 4](#_Toc92385180)

[1.1 Định nghĩa: 4](#_Toc92385181)

[2 Các phép toán trên BST 4](#_Toc92385182)

[2.1.Tìm kiếm 4](#_Toc92385183)

[2.1.1.Thuật toán: 4](#_Toc92385184)

[2.2.Thêm node vào BST 4](#_Toc92385185)

[2.2.1. Thuật toán 4](#_Toc92385186)

[2.2.2. Các bước thực hiện 4](#_Toc92385187)

[2.3. Xóa node trên BST 5](#_Toc92385188)

[2.3.1. Thuật toán 5](#_Toc92385189)

[2.3.2. Các bước thực hiện 5](#_Toc92385193)

[CHƯƠNG 2 : Bài toán cây nhị phân tìm kiếm 6](#_Toc92385194)

[2.1. Đặc tả bài toán 6](#_Toc92385195)

[2. 2. Yêu cầu hệ thống 6](#_Toc92385196)

[2.3. Phân tích thiết kế chương trình 6](#_Toc92385198)

[2.3.1. Cấu trúc chương trình 6](#_Toc92385199)

[2.3.2. Các chức năng của chương trình 14](#_Toc92385200)

[2.4. Cài đặt chương trình 14](#_Toc92385201)

[CHƯƠNG 3: Kết luận 19](#_Toc92385202)

[1.Các kết quả đạt được 19](#_Toc92385203)

[2.Ưu và nhược điểm của đề tài 19](#_Toc92385204)

[3. Hướng phát triển 19](#_Toc92385205)

[4. Tài liệu tham khảo 19](#_Toc92385207)

# PHẦN MỞ ĐẦU

1. **Lý do chọn đề tài:**

**Công nghệ thông tin** là một trong những ngành phát triển vượt bậc trong những năm gần đây. Ngày nay với sự phát triển nhanh chóng của xã hội thì công nghệ thông tin được ứng dụng rộng rãi ở hầu hết tất cả các lĩnh vực và ngày càng đóng vai trò quan trọng, trở thành một phần thiết yếu trong đời sống hằng ngày. Công nghệ thông tin là một ngành đòi hỏi ở người học một nền tảng kiến thức vững chắc, sự tư duy logic cao, hiểu biết sâu rộng trên nhiều lĩnh vực. Với chúng em hiện đang là những sinh viên công nghệ thông tin cần phải có sự đầu tư, không ngừng học hỏi để nâng cao kiến thức. Do đó để củng cố lại kiến thức đã học, đề tài mà em chọn để thực hiện là **thao tác trên cây nhị phân tìm kiếm**

1. **Mục tiêu của đề tài.**
   * Củng cố lại kiến thức đã học về cấu trúc struct, con trỏ. ôn tập lại kiến thức về các thao tác lập trình cơ bản.
   * Rèn luyện kỹ năng lập trình trên ngôn ngữ C++.
   * Ứng dụng lý thuyết đã học giải quyết bài toán ứng dụng cụ thể
2. **Phạm vi nghiên cứu.**
   * Lý thuyết về cây nhị phân tìm kiếm (BST).
   * Bài toán thực hiện các thao các cơ bản như thêm node, tìm kiếm node và xóa node.
3. **Phương pháp nghiên cứu.**
   * Nghiên cứu giáo trình cấu trúc dữ liệu và giải thuật, kĩ thuật đồ họa.
   * Tìm kiếm và nghiên cứu trên mạng Internet.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.Lý thuyết về cây tìm kiếm nhị phân:

### 1.1 Định nghĩa:

 Là một cây nhị phân và có thêm các ràng buộc sau đây:

- Giá trị của tất cả các Node ở cây con bên trái phải < giá trị của Node gốc.

- Giá trị của tất cả các Node ở cây con bên phải phải > giá trị của Node gốc.

- Tất cả các cây con(bao gồm bên trái và phải) cũng đều phải đảm bảo 2 tính chất trên.

## 2 Các phép toán trên BST

### 2.1.Tìm kiếm

### 2.1.1.Thuật toán:

Việc tìm một khóa trên BST thực hiện nhờ đệ quy. Chúng ta bắt đầu từ gốc. Nếu node cần tìm bằng node của gốc thì tìm thấy, nếu node cần tìm nhỏ hơn node ở gốc, ta phải tìm nó trên cây con trái, nếu khóa cần tìm lớn hơn node ở gốc, ta phải tìm nó trên cây con phải. Nếu cây con (trái hoặc phải) là rỗng thì node cần tìm không có trên cây.

**2.1.2. Các bước thực hiện:**

* B1: Nếu Node hiện tại có giá trị = giá trị cần tìm, trả về true và kết thúc.
* B2: Nếu Node hiện tại có giá trị > giá trị cần tìm, gọi đệ quy tìm ở cây con bên trái.
* B3: Nếu Node hiện tại có giá trị < giá trị cần tìm, gọi đệ quy tìm ở cây con bên phải
* B4: Nếu tìm đến hết cây(Node đó = NULL) mà không xảy ra (1), trả về false và kết thúc.

## 2.2.Thêm node vào BST

### 2.2.1. Thuật toán

Việc thêm 1 node vào cây nhị phân phân tìm kiếm phải thỏa mãn điều kiện như đã đề cập ở trên, nếu node cần thêm nhỏ hơn node gốc thì đệ quy sang trái node gốc, đệ quy đến khi thỏa mãn điều kiện của một cây nhị phân tìm kiếm thì tiến hành thêm node. Và ngược lại, nếu node cần thêm lớn hơn node gốc thì đệ quy sang phải node gốc, đệ quy đến khi thỏa mãn điều kiện của một cây nhị phân tìm kiếm thì tiến hành thêm node.

### 2.2.2. Các bước thực hiện

* B1: Nếu Node hiện tại = NULL, đó là vị trí cần thêm. Thêm vào BST và kết thúc
* B2: Nếu giá trị cần thêm < giá trị node hiện tại, gọi đệ quy hàm thêm vào cây con bên trái
* B3: Nếu giá trị cần thêm > giá trị node hiện tại, gọi đệ quy hàm thêm vào cây con bên phải.

## 2.3. Xóa node trên BST

### 2.3.1. Thuật toán

### Để xóa một node trước hết ta phải thực hiện tìm nút cần xóa, sau khi tìm thấy nút cần xóa.

### Nếu node cần xóa là node có 1 con hoặc node lá thì ta chỉ việc xóa node đó.

### Nếu node cần xóa là node có 2 con thì ta cần phải làm tìm một node thế mạng để thay node cần xóa rồi xóa node thế mạng đó.

### 2.3.2. Các bước thực hiện

* Bước 1: Duyệt đệ quy tìm node cần xóa.
* Bước 2:
  + Nếu node cần xóa là node lá hoặc node có 1 con thì thực hiện xóa
  + Nếu node cần xóa là node có hai con thì:

1. Tìm node thế mạng bằng cách duyệt sang node bên phải cùng của cây con trái hay node bên trái cùng của cây con phải của node cần xóa
2. Cập nhật giá trị của node cần xóa bằng giá trị node vừa tìm được.
3. Thực hiện xóa node vừa tìm được ở bước 1

# CHƯƠNG 2 : Bài toán cây nhị phân tìm kiếm

## 2.1. Đặc tả bài toán

Cây nhị phân được sử dụng vào nhiều mục đích khác nhau. Tuy nhiên việc sử dụng cây nhị phân để lưu giữ và tìm kiếm thông tin vẫn là một trong những áp dụng quan trọng nhất của cây nhị phân.

## 2. Yêu cầu hệ thống

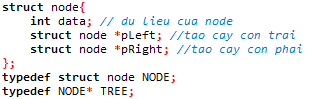
## Việc ứng dụng cây nhị phân vào đời sống giúp việc quản lí và tìm kiếm thông tin dễ dàng hơn.

Chương trình có thể được viết bằng nhiều ngôn ngữ lập trình khác nhau như C, C++, Python, Java... Nhưng trong bài toán này, ngôn ngữ đc sử dụng là C++.

## 2.3. Phân tích thiết kế chương trình

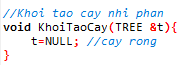
### 2.3.1. Cấu trúc chương trình

Ta dùng kiểu struct để lưu trữ các dữ liệu cần thiết của một cây BST

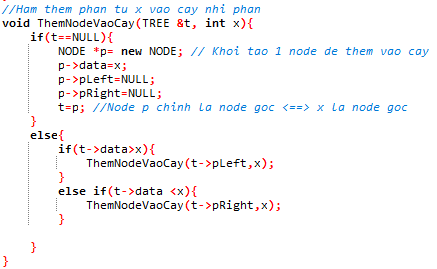


Tiếp theo để có được cây thì ta phải có 1 hàm khởi tạo cây

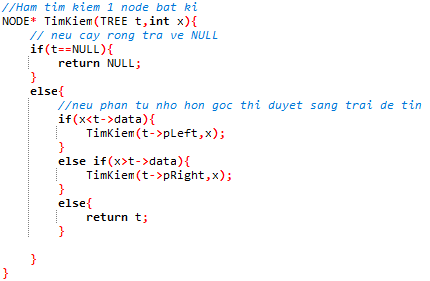
* **Hàm khởi tạo cây**



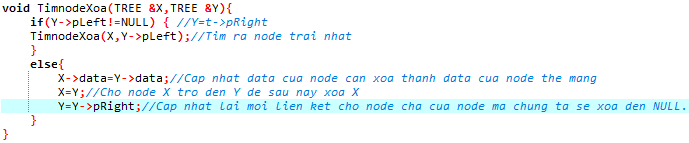
* **Hàm chuyển thêm node vào cây:**



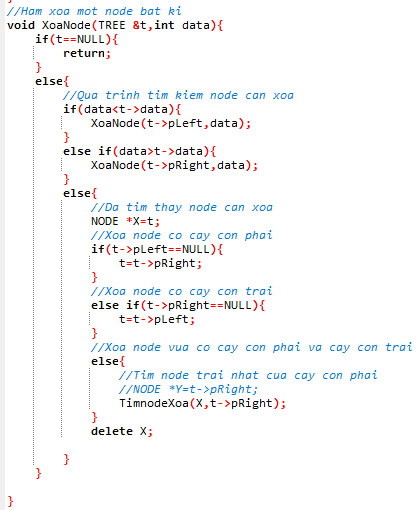
* **Hàm tìm kiếm 1 node trên cây**



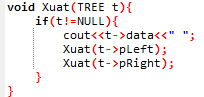
* **Hàm tìm node thế mạng để sử dụng trong việc xóa node có 2 con**



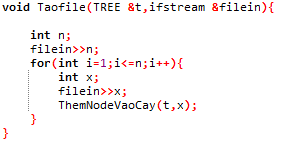
* **Hàm xóa node cần xóa**



* **Hàm duyệt cây**

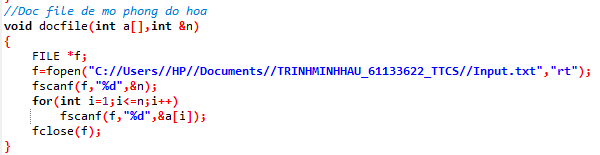


* **Hàm tạo file**

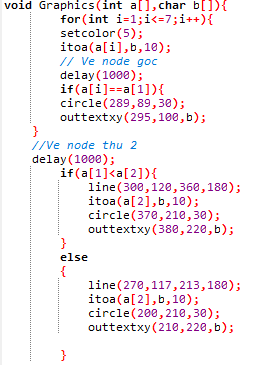
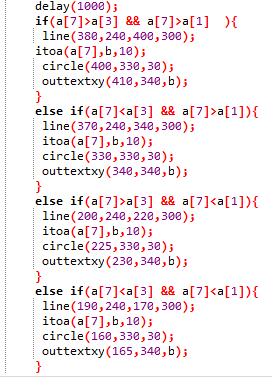
Hàm này sẽ cho phép ta nhập dữ liệu từ file

* **Hàm đọc file vào mảng**

Hàm này sẽ giúp việc minh họa đồ họa từ file

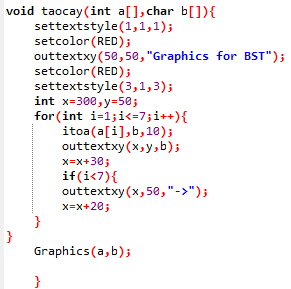


* **Hàm mô phỏng việc chèn thêm một node vào cây**

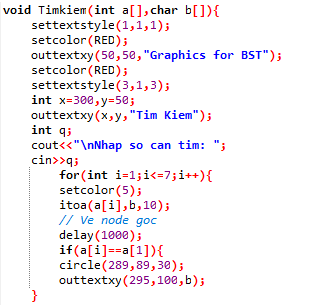
 

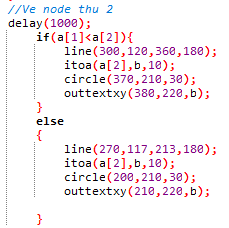
* **Hàm mô phỏng thêm node vào cây**

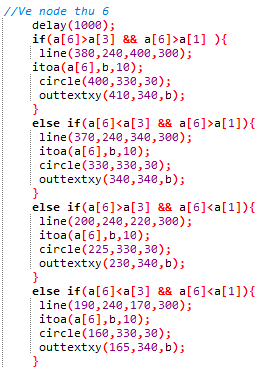
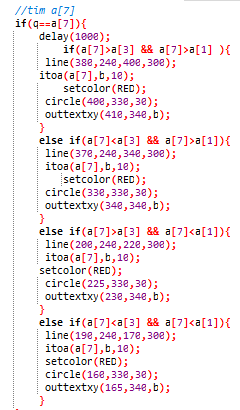
Hàm này dựa theo hàm Graphis() đề cập ở trên



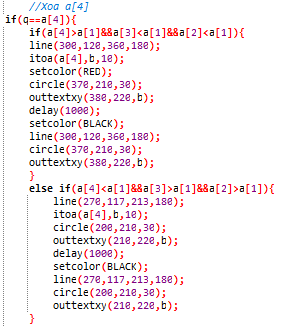
* **Hàm mô phỏng việc tìm kiếm 1 node trên cây**

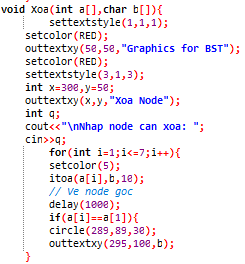


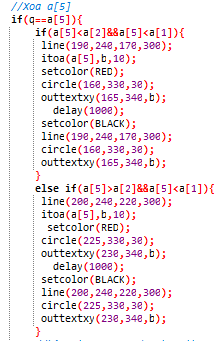
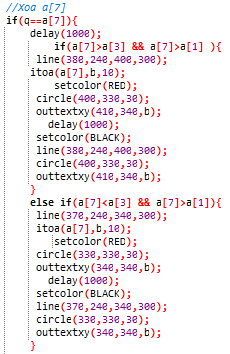




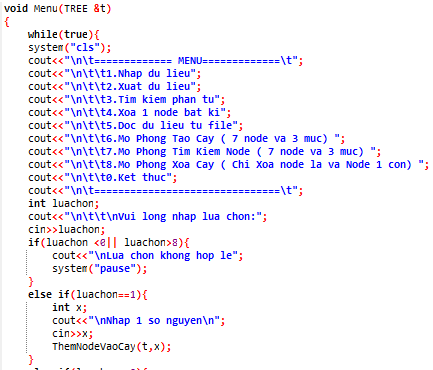
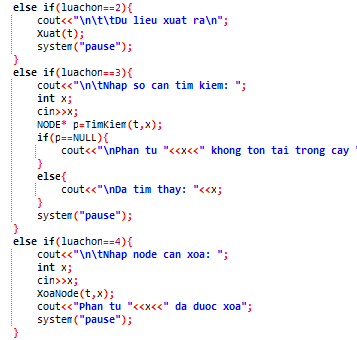
* **Hàm mô phỏng việc xóa node**
* **Hàm xóa node lá hoặc node có 1 con**

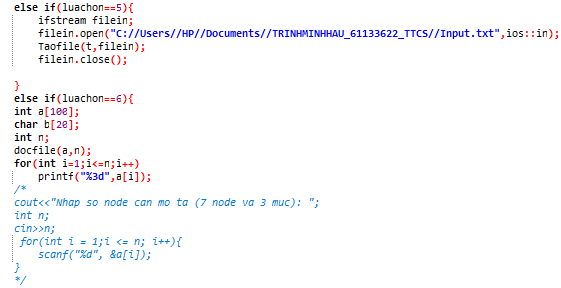


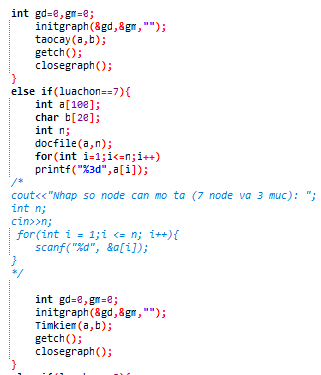


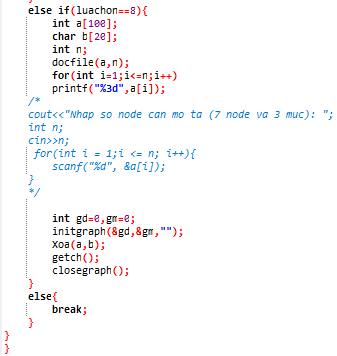


* **Hàm chứa các chức năng chính**

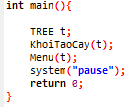








* **Hàm main**

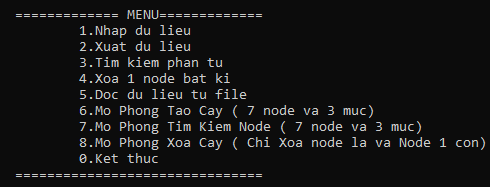


### 2.3.2. Các chức năng của chương trình

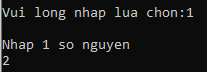
* Nhập các node từ bàn phím hoặc từ file
* Chọn chức năng cần thực hiện từ menu
* Xuất ra kết quả của các chức năng ra màn hình

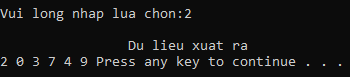
## 2.4. Cài đặt chương trình

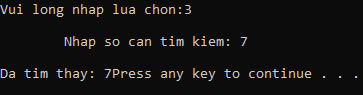
* Menu chức năng



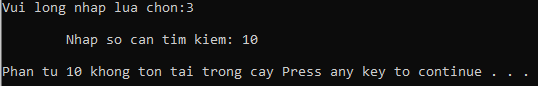
* Nhập dữ liệu

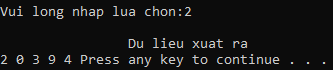
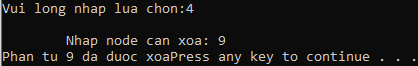
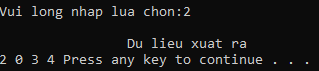


* Xuất dữ liệu
* Tìm kiếm phần tử
* Nếu có phần tử



* Nếu phần tử không tồn tại

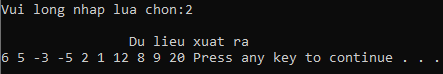


* Xóa 1 node bất kì
* Hiển thị xem có phần tử nào
*  Xóa node số 9
* Kiểm tra lại
* Đọc dữ liệu từ file

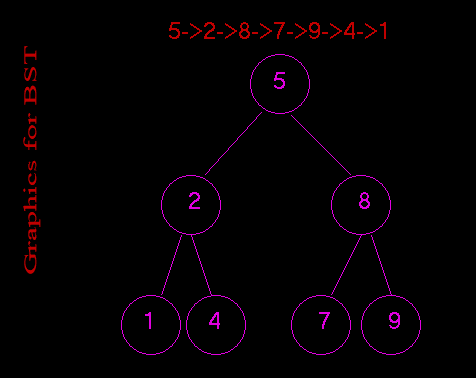
- File chuẩn bị sẵn



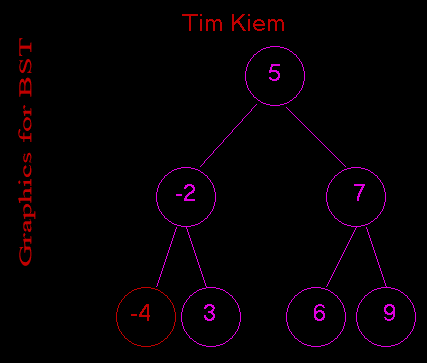
- Dữ liệu được đọc từ file

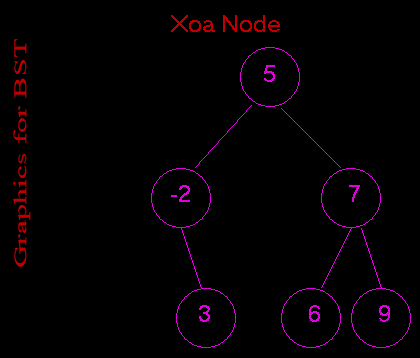


* Mô phỏng thêm dữ liệu vào cây



* Mô phỏng tìm kiếm dữ liệu



* Mô phỏng xóa node:
* Xóa node lá hoặc node có 1 con
* 

# CHƯƠNG 3: Kết luận

1. **Các kết quả đạt được**
   * Trình bày khái quát thuật toán BST
   * Xây dựng được các chức năng cơ bản trên cây nhị phân tìm kiếm (BST)
   * Mô phỏng được các chức năng cơ bản bằng đồ họa
2. **Ưu và nhược điểm của đề tài**

* Ưu điểm
  + Chương trình dễ sử dụng, giao diện thân thiện
* Nhược điểm:
  + Giao diện đồ họa chỉ ở mức mô phỏng được 7 node mức 3, chưa đi vào tổng quát.
  + Code chương trình còn dài, chưa tối ưu

1. **Hướng phát triển**
   * Xây dựng được mô phỏng đồ họa ở mức tổng quát

1. **Tài liệu tham khảo**

* Giáo trình cấu trúc dữ liệu và giải thuật – Nguyễn Đức Thuần - ĐH Nha Trang
* Giáo trình kỹ thuật đồ họa – Đoàn Vũ Thịnh – ĐH Nha Trang