

TP. Hồ Chí Minh, ngày 02 tháng 12 năm 2021

**Giảng viên hướng dẫn: Th.S Lê Vĩnh Thịnh**

**Sinh viên thực hiện:**

**Hoàng Vũ Trường Giang - 20110237**

**Nguyễn Như Tú - 20110250**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**ĐỀ TÀI: DEMO THUẬT TOÁN CÂY TRONG DATA STRUCTURE**

**MỤC LỤC**

MỤC LỤC………………………………………………………………………………2

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 3](#_Toc90369966)

[**1.1. Lời nói đầu** 3](#_Toc90369967)

[**1.2. Phân công nhiệm vụ** 4](#_Toc90369968)

[**1.3. Lí do chọn đề tài** 5](#_Toc90369969)

[**1.4. Giới thiệu về đề tài** 6](#_Toc90369970)

[**1.4.1. Cấu trúc cây** 6](#_Toc90369971)

[**1.4.2. Cây nhị phân** 6](#_Toc90369972)

[**1.4.3. Cây tìm kiếm nhị phân** 7](#_Toc90369973)

[CHƯƠNG 2. GIAO DIỆN, TÍNH NĂNG VÀ CÁCH THỨC HOẠT ĐỘNG 9](#_Toc90369974)

[**2.1. Giao diện khi khởi động** 9](#_Toc90369975)

[**2.2. Tính năng chính và cách thức hoạt động** 9](#_Toc90369976)

[**2.2.1. Nhập dữ liệu từ bàn phím (Insert)** 9](#_Toc90369977)

[**2.2.2. Tính năng tìm kiếm (Find)** 11](#_Toc90369978)

[**2.2.3. Tính năng xóa nút (Delete)** 12](#_Toc90369979)

[**2.2.4. Tìm số lớn nhất và số nhỏ nhất (Max và Min)** 12](#_Toc90369980)

[**2.2.5. Tạo mảng (Display)** 13](#_Toc90369981)

[**2.2.6. Xóa cây (Reset)** 13](#_Toc90369982)

[CHƯƠNG 3. MÔ TẢ CHI TIẾT 14](#_Toc90369983)

[**3.1. Cấu trúc chương trình** 14](#_Toc90369984)

[**3.1.1. Diagram** 14](#_Toc90369985)

[**3.1.2. Thiết kế giao diện** 14](#_Toc90369986)

[**3.2. Các tính năng quản lí trên Node** 15](#_Toc90369987)

[**3.2.1. Class Node** 15](#_Toc90369988)

[**3.2.2. Một số code thực thi** 16](#_Toc90369989)

[**3.3. Các tính năng quản lí trên Tree** 18](#_Toc90369990)

[**3.3.1. Class Tree** 18](#_Toc90369991)

[**3.3.2. Một số code thực thi** 18](#_Toc90369992)

[TỔNG KẾT 20](#_Toc90369993)

1. **TỔNG QUAN ĐỀ TÀI**
   1. **Lời nói đầu**

Lời đầu tiên, nhóm 11 chúng em xin trân trọng cảm ơn và gửi lời tri ân đến ThS. Lê Vĩnh Thịnh, Thầy là người trực tiếp chỉ bảo, giảng dạy nhóm trong quá trình học tập môn Lập trình Hướng đối tượng cũng như tận tình giúp đỡ, hỗ trợ chúng em để có thể hoàn thành tốt nhất có thể đồ án cuối kỳ môn học này.

Trong suốt quá trình hoạt động nghiêm túc, nhóm đã lần lượt nghiên cứu và thực hiện các nội dung có trong đề tài. Tuy nhiên, vì sự giới hạn về kiến thức cũng như khả năng lập luận của các thành viên nhóm còn nhiều điểm hạn chế dẫn đến việc khó tránh khỏi còn gặp ít nhiều những thiếu xót. Chúng em rất hi vọng nhận được sự nhận xét, góp ý và chỉ dẫn của Thầy để bài báo cáo cũng như đề tài của chúng em được ngày càng hoàn thiện hơn ạ.

Một lần nữa, nhóm xin gửi lời chân thành cảm ơn Thầy.

*Nhóm sinh viên.*

* 1. **Phân công nhiệm vụ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và Tên** | **MSSV** | **Nhiệm vụ** | **Tiến độ hoàn thành** |
| Hoàng Vũ Trường Giang | 20110237 | Nghiên cứu đề tài  Viết code ứng dụng  Áp dụng thuật toán  Làm slide báo cáo | 100% |
| Nguyễn Như Tú | 20110250 | Nghiên cứu đề tài  Viết code ứng dụng  Xây dựng giao diện Viết báo cáo Word | 100% |

* 1. **Lí do chọn đề tài**

Hiện nay, cấu trúc cây đặt biệt là cây tìm kiếm nhị phân trở nên khá phổ biến áp dụng trong tính toán, đo đạc cũng như một số lĩnh vực khác của đời sống, đòi hỏi chúng ta phải có công cụ hỗ trợ nâng cao hiệu quả và hiệu suất khi tra cứu, sắp xếp các con số. Đây là một dạng cấu trúc được sử dụng rất nhiều trong tìm kiếm, nó được tối ưu nhất trong các cấu trúc dữ liệu mà mình đã giới thiệu. Hơn nữa, công cụ còn phải đáp ứng được nhu cầu thao tác trên một dãy số cực kì nhanh chóng, đơn giản và dễ dàng sử dụng. Trong quá trình nghiên cứu đồ án của môn học OOP, nhóm nhận thấy đây là một đề tài tiềm năng và hữu ích để khai thác. Qua đó, nhóm còn áp dụng các kiến thức đã được học ở môn học này và Cấu trúc dữ liệu vào để xây dựng nên một công cụ hiệu quả nhất có thể. Chính vì thế, nhóm quyết định xây dựng ứng dụng Demo thuật toán cây trong DataStucture mang tên “BINARY SEARCH TREE” là giải pháp hữu hiệu giải quyết các vấn đề nêu trên. Trong quá trình hoàn thiện sản phẩm của mình, nhóm có tham khảo thêm các tài liệu và công cụ liên quan trên Internet.

* 1. **Giới thiệu về đề tài**
     1. **Cấu trúc cây**

Diagram, shape

Description automatically generatedTrong khoa học máy tính, cây (Tree) là một cấu trúc dữ liệu được sử dụng rộng rãi gồm một tập hợp các nút (Node) được liên kết với nhau theo quan hệ cha con. Cây trong cấu trúc dữ liệu đầu tiên là mô phỏng (hay nói cách khác là sự sao chép) của cây (có gốc) trong lý thuyết đồ thị. Hầu như mọi khái niệm trong cây của lý thuyết đồ thị đều được thể hiện trong cấu trúc dữ liệu. Tuy nhiên, cây trong cấu trúc dữ liệu đã tìm được ứng dụng phong phú và hiệu quả hơn trong nhiều giải thuật. Khi phân tích các giải thuật trên cấu trúc dữ liệu cây, người ta vẫn thường vẽ ra các cây tương ứng trong lý thuyết đồ thị.

Ưu điểm của cấy trúc dữ liệu cây so với các cấu trúc khác là:

* Phân cấp dữ liệu.
* Tìm kiếm dễ dàng hơn.
* Thao tác trên các danh sách dữ liệu đã sắp xếp.

Trong cấu trúc dữ liệu cây, có hai dạng cấu trúc chính đó là cấu trúc cây nhị phân và cấu trúc cây nhị phân tìm kiếm.

* + 1. **Cây nhị phân**

Cây nhị phân là một cấu trúc dữ liệu đặc biệt được sử dụng cho mục đích lưu trữ dữ liệu. Một cây nhị phân có một điều kiện đặc biệt là mỗi nút có thể có tối đa hai nút con. Một cây nhị phân tận dụng lợi thế của hai kiểu cấu trúc dữ liệu: Một mảng đã sắp thứ tự và một danh sách liên kết (Linked List), do đó việc tìm kiếm sẽ nhanh như trong mảng đã sắp thứ tự và các thao tác chèn và xóa cũng sẽ nhanh bằng trong Linked List.

Các thành phần cơ bản của cây nhị phân bao gồm:

* Root: Được gọi là Node gốc của cây (là một Node cha), một cây chỉ có một Node gốc duy nhất và nó không có Node cha nào.
* Parent Node: Là Node cha của một Node cụ thể nào đó.
* Child Node: Là Node con của một Node cụ thể nào đó.
* Sub-tree: Là cây con biểu diển các con của một Node.
* LeafNode: Là Node không có Node con.
* Siblings: Các Node có cùng một cha.
* Internal Node: Node có ít nhất một Node con.
* External Node: Node không có Node con nào.

Mỗi Node bao gồm 3 thành phần:

* Data: Giá trị của một phần tử
* Left pointer: Con trỏ trỏ đến cây nhị phân bên trái Node.
* Right pointer: Con trỏ trỏ đến cây nhị phân bên phải Node.
  + 1. **Cây tìm kiếm nhị phân**

Cây tìm kiếm nhị phân là một cấu trúc dữ liệu cơ bản được sử dụng để xây dựng các cấu trúc dữ liệu trừu tượng hơn như các tập hợp, đa tập hợp, các dãy kết hợp. Nếu một cây tìm kiếm nhị phân có chứa các giá trị giống nhau thì nó biểu diễn một đa tập hợp. Cây loại này sử dụng các bất đẳng thức không nghiêm ngặt. Mọi nút trong cây con trái có khóa nhỏ hơn khóa của nút cha, mọi nút trên cây con phải có nút lớn hơn hoặc bằng khóa của nút cha.

Nếu một cây tìm kiếm nhị phân không chứa các giá trị giống nhau thì nó biểu diễn một tập hợp đơn trị như trong lý thuyết tập hợp. Cây loại này sử dụng các bất đẳng thức nghiêm ngặt. Mọi nút trong cây con trái có khóa nhỏ hơn khóa của nút cha, mọi nút trên cây con phải có khóa lớn hơn khóa của nút cha. Việc chọn đưa các giá trị bằng nhau vào cây con phải (hay trái) là tùy theo mỗi người. Một số người cũng đưa các giá trị bằng nhau vào cả hai phía, nhưng khi đó việc tìm kiếm trở nên phức tạp Shape

Description automatically generated with low confidencehơn.

Trong cây nhị phân tìm kiếm ta có thể thực hiện các thao tác sau:

* Chèn một phần tử vào trong một cây.
* Tìm kiếm phần tử trong cây.
* Duyệt cây.
* Đo chiều cao của cây.

Trên đây là các thao tác thường được sử dụng nhiều trong cây. Đặc biệt là tìm kiếm phần tử trong cây, như cái tên của nó là cây nhị phân tìm kiếm. Đây là một cấu trúc dữ liệu được sử dụng trong các bài toán tìm kiếm rất nhiều, bởi tính chính xác và tốc độ của nó.

1. **GIAO DIỆN, TÍNH NĂNG VÀ CÁCH THỨC HOẠT ĐỘNG**
   1. **Giao diện khi khởi động**

Sau khi khởi động, một cửa sổ Winform hiện ra cho thấy giao diện chính và một số tính năng cơ bản. Công cụ được phát triển có tên là “BINARY SEARCH TREE”, khoảng trống ở giữa màn hình là nơi mà xuất hiện các phần của một cây tìm kiếm nhị phân sau khi thao tác. Bên phải là các nút thể hiện cho các tính năng chính của công cụ này. Bên cạnh các nút chính là những ô trống cho phép người dùng nhập từ bàn phím các con số phù hợp với mục tiêu mà chúng ta cần bộ công cụ thực hiện.

Graphical user interface, text, application

Description automatically generated

* 1. **Tính năng chính và cách thức hoạt động**
     1. **Nhập dữ liệu từ bàn phím (Insert)**

Người dùng thực hiện việc nhập số từ bàn phím vào ô trống, sau đó nhấn lệnh Insert để gán số trở thành một Node trên cây. Số được nhập đầu tiên là Node gốc của cây (hay còn được gọi là Root). Lưu ý: chỉ nhập số thỏa mãn yêu cầu của một cây tìm kiếm nhị phân (số nguyên), nếu không màn hình sẽ hiện cảnh báo lỗi.

Graphical user interface, application

Description automatically generatedChart, line chart

Description automatically generatedSau đó, tiếp tục nhập những con số khác vào ô trống và các Node sẽ được thể hiện trên cây tuân theo quy tắc của một cây nhị phân tìm kiếm đó là mọi nút trong cây con trái có khóa nhỏ hơn khóa của nút cha, mọi nút trên cây con phải có nút lớn hơn nút cha. Giữa nút cha và con là đường liên kết có dạng màu tím như hình.

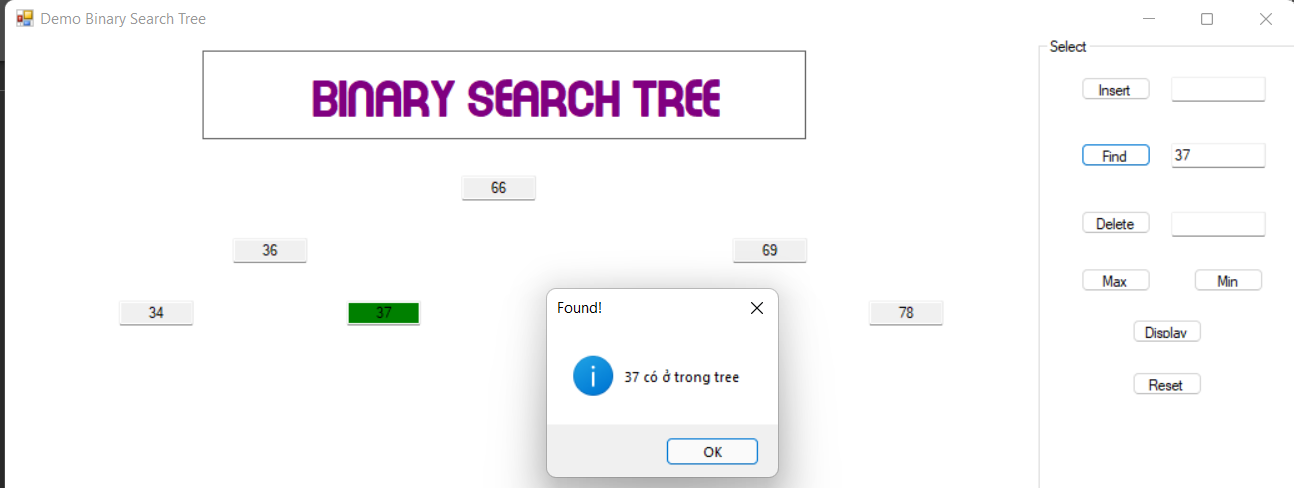
Graphical user interface, application

Description automatically generatedNếu người dùng nhập một con số trùng với các số đã xuất hiện trên cây, sẽ có một bảng cảnh báo hiện lên thông báo rằng con số đó có giá trị trùng lặp và chúng ta phải tiến hành nhập vào một số khác.

Graphical user interface, text, application

Description automatically generatedNếu người dùng nhập vào các chữ số không thỏa mãn, ví dụ số thập phân, phân số, ký tự chữ, ký tự đặc biệt,… màn hình sẽ hiện cảnh báo lỗi yêu cầu bạn nhập vào một số khác thỏa mãn yêu cầu cơ bản của một cây tìm kiếm nhị phân.

* + 1. **Tính năng tìm kiếm (Find)**

****Đây là tính năng cho phép tìm kiếm các Node có trong cây tìm kiếm nhị phân. Nhập từ bàn phím con số cần tìm, sau đó nhấn nút “Find” để tiến hành tìm kiếm. Nếu có, màn hình sẽ hiện bảng thông báo và Node chứa số cần tìm đó được tô xanh lên.

Graphical user interface, text, application

Description automatically generatedNếu không, sẽ hiện thông báo số cần tìm không có trong cây.

* + 1. **Tính năng xóa nút (Delete)**

Diagram

Description automatically generated with medium confidenceA picture containing chart

Description automatically generatedĐể xóa một Node trên cây, người dùng tiến hành nhập con số mình muốn xóa vào ô trống, sau đó nhấn “Delete”. Nếu nhập một số mà trên trên cây tìm kiếm nhị phân đã tạo không có, màn hình sẽ hiện thông báo rằng số cần xóa không có trên cây.

* + 1. **Tìm số lớn nhất và số nhỏ nhất (Max và Min)**

Graphical user interface, application

Description automatically generatedĐể tìm một Node lớn nhất trong cây đã tạo, người dùng tiến hành nhấn nút “Max”, Node lớn nhất sẽ được tô xanh đồng thời màn hình hiện thông báo cho biết giá trị lớn nhất. Tương tự các bước để tìm một Node bé nhất trong cây.

Graphical user interface, application

Description automatically generated

* + 1. **Tạo mảng (Display)**

Graphical user interface, application

Description automatically generatedKhi muốn thiết kế một mảng các số xuất hiện trong cây tìm kiếm nhị phân đã tạo sắp xếp theo thứ tự tăng dần, người dùng nhấm vào nút “Display”.

* + 1. **Xóa cây (Reset)**

**Graphical user interface, application

Description automatically generated**Để xóa hoàn toàn một cây đã được tạo ra hay đưa giao diện trở về như ban đầu, người dùng nhấn phím “Reset”.

1. **MÔ TẢ CHI TIẾT**
   1. **Cấu trúc chương trình**
      1. **Graphical user interface, text, application

         Description automatically generatedDiagram**

Bao gồm 2 class mang chức năng quản lý là Node và Tree. Hai class này có chức năng riêng biệt chuyên xử lí các thao tác với Node trên cây và thao tác trên toàn bộ cây.

* + 1. **Thiết kế giao diện**

**A picture containing text, screenshot, monitor, screen

Description automatically generated**Đây là giao diện khi thiết kế của công cụ được thiết kế với tiêu chí rõ ràng và dễ dàng thao tác trên từng mục.

* 1. **Các tính năng quản lí trên Node**
     1. **Class Node**

|  |  |
| --- | --- |
| **Thuộc tính** | |
| public int Data | |
| public Node Left | |
| public Node Right | |
| public bool IsLeftChild | |
| public Node Parent | |
| public TextBox Box | |
| **Phương thức** | |
| private int GetNumberOfParent() | Trả về số cha của node hiện tại. |
| public bool isLeftOfRoot() | Cho biết node hiện tại có nằm bên trái của Root không. |
| public void Insert(Node nodeInsert) | Chèn một node. |
| public Node Find(int value) | Biểu diễn các node theo một mảng tăng dần. |
| public void Review() | Thiết lập vị trí phù hợp của Box. |
| public void Show() | Biểu diễn lại cây khi có sự thay đổi. |
| public Node GetMax() | Trả về giá trị lớn nhất của cây có root là node hiện tại. |
| public Node GetMin() | Trả về giá trị nhỏ nhất của cây có root là node hiện tại. |
| public void Reset() | Thực hiện xóa cây có root là node hiện tại. |

* + 1. **Text

       Description automatically generatedText

       Description automatically generatedText

       Description automatically generatedMột số code thực thi**

**Text

Description automatically generatedText

Description automatically generated**

* 1. **Các tính năng quản lí trên Tree**
     1. **Class Tree**

|  |  |
| --- | --- |
| **Thuộc tính** | |
| public Node Root |  |
| **Phương thức** | |
| public void Insert(Node nodeInsert) | Chèn một node vào cây. |
| public Node Find(int value) | Tìm xem node có tồn tại trên cây không. |
| public string Display() | In thành mảng theo thứ tự tăng dần. |
| public Node GetMax() | Tìm giá trị lớn nhất của cây. |
| public Node GetMin() | Tìm giá trị nhỏ nhất của cây. |
| public bool Remove(int value) | Xóa một node trên cây. |
| public Node GetSuccessor(Node node) | Tìm node kế thừa của node bị xóa. |
| public void Reset() | Xóa toàn bộ cây. |

* + 1. **Text

       Description automatically generatedMột số code thực thi**

**Text

Description automatically generatedText

Description automatically generated**

**TỔNG KẾT**

Sau thời gian nghiên cứu và thực hiện nghiêm túc đề tài, nhóm đã phát triển được công cụ giúp “Demo các thuật toán trên cây tìm kiếm nhị phân”. Trong quá trình thực hiện, vì còn một số rào cản về mặt kiến thức cũng như công nghệ mới nên việc tồn tại một vài hạn chế là điều khó tránh khỏi. Để khắc phục được những hạn chế này, chúng em sẽ phải cố gắng tìm tòi các công nghệ mới, nắm chắc kiến thức của môn học OOP hơn nữa và rất mong nhận được sự chỉ bảo cũng như đóng góp của thầy để giúp chúng em phát triển hơn trong tương lai. Một lần nữa xin gửi lời cảm ơn chân thành đến thầy, chúc thầy sức khỏe và đạt được nhiều thành công trong cuộc sống.

**Trân trọng ./.**