

BÁO CÁO LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG  
MÃ LỚP 143577  
NHÓM 30

Link project: <https://github.com/ngnlesn33/OOLT.VN.20231-30.git>

## Contents

1. Đề bài	1
2. Yêu cầu bài toán	1
<b>3. Biểu đồ</b>	<b>4</b>
4. Mô tả các loại cây và giải thuật	5
4.1 Generic tre:	5
4.2 Binary Search Tree (BST)	6
4.3 Adelson-Velskii Landis (AVL)	7
4.4 Red-Black Tree	7
<b>5. Phân công công việc</b>	<b>8</b>

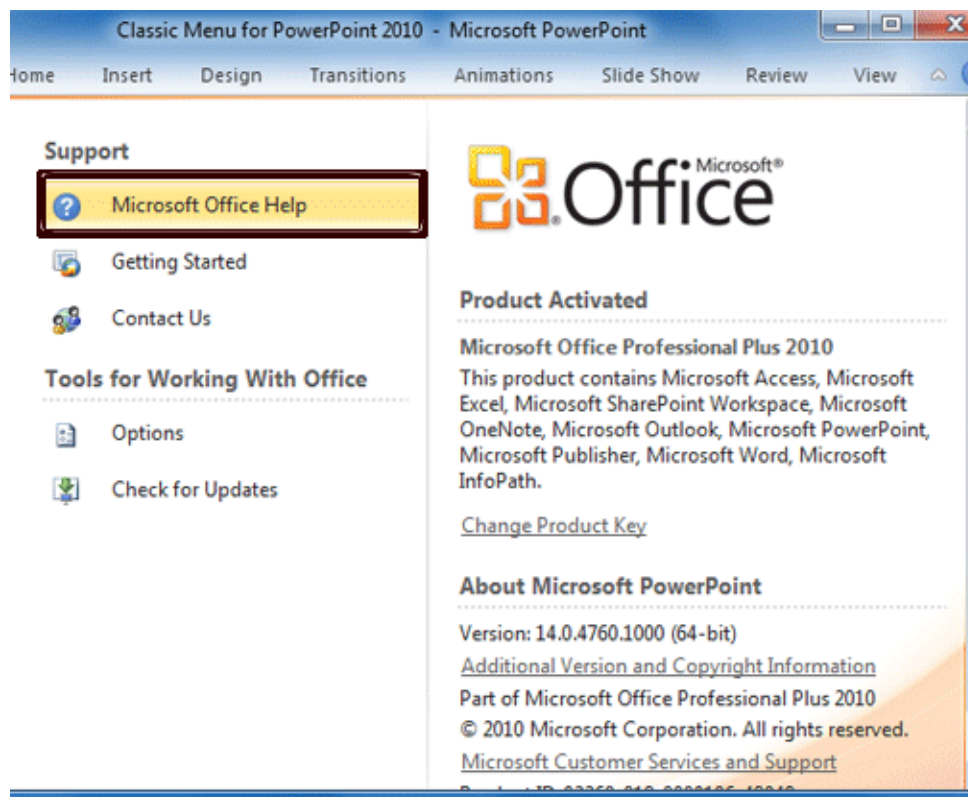
## 1. Đề bài

Visualization of operations on tree data structures

## 2. Yêu cầu bài toán

- **Tổng quan:** Cây là một cấu trúc dữ liệu hữu ích với rất nhiều ứng dụng trong khoa học máy tính. Trong dự án này, bạn sẽ thiết kế một chương trình để hiển thị và giải thích một số thao tác cơ bản của bốn loại cấu trúc cây.
  - o Generic Tree
  - o Binary Search Tree (BST)
  - o Adelson-Velskii Landis (AVL)
  - o Balanced tree
- Giao diện: Bạn có thể tham khảo nguồn này để có một số ý tưởng:  
<https://visualgo.net/en/bst>
- Thiết kế:
  - o Chỉ xem xét các cây **vô hướng có trọng số**, với **các nút có giá trị nguyên** và **không cho phép các giá trị nút bị trùng lặp**.
    - Vô hướng: Các cạnh trong cây không có hướng. Điều này có nghĩa là nếu có một cạnh giữa các nút A và B, bạn có thể di chuyển từ A đến B và từ B đến A.

- Có trọng số: Mỗi cạnh trong cây có một trọng số (hoặc chi phí) liên quan đến nó. Trọng số này có thể đại diện cho nhiều thứ tùy thuộc vào vấn đề đang được giải quyết, chẳng hạn như khoảng cách giữa hai thành phố, chi phí của một giao dịch, v.v.
  - Giá trị Nút là số nguyên: Mỗi nút trong cây có một giá trị số nguyên.
  - Không có giá trị nút trùng lặp: Mỗi nút trong cây có một giá trị số nguyên duy nhất. Điều này có nghĩa là không có hai nút nào chia sẻ cùng một giá trị.
- o Đối với cây cân bằng và cây nhị phân cân bằng, khoảng cách lớn nhất từ gốc đến các nút lá phải được **chọn bởi người dùng**.
  - o **Trên menu chính:** tiêu đề, thanh điều hướng để người dùng chọn giữa bốn loại cây, menu trợ giúp và thoát.
    - Bắt buộc người dùng phải chọn loại cây
    - Menu trợ giúp: cung cấp cho người dùng hướng dẫn cách thao tác cơ bản và mục đích của dự án. Chắc hơi giống như này:



- Nút thoát thoát khỏi ứng dụng. Nhớ yêu cầu xác nhận.
- o Các thao tác trong phần mô hình hóa cây:

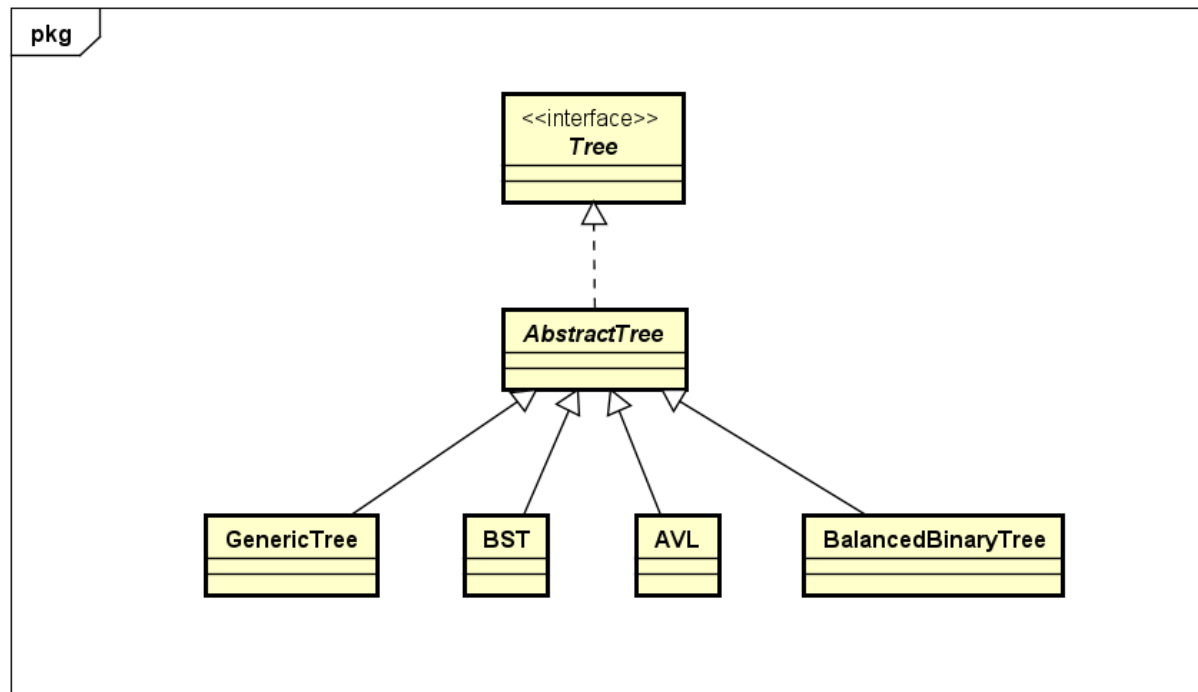
- Người dùng có thể chọn để trực quan hóa một trong sáu thao tác, bằng cách chọn một tùy chọn trong menu thao tác, sau đó cung cấp các tham số cần thiết.

Thao tác	Tham số	Mô tả
Tạo	Không có	Tạo một cây mới
Thêm	Giá trị của node cha và giá trị của node muốn thêm	Thêm nút mới với giá trị được chỉ định làm nút con của nút cha được chỉ định
Xóa	Giá trị của node muốn xóa	Xóa nút khỏi cây
Sửa	Giá trị hiện tại của Node và giá trị mới	Thay đổi nút có giá trị hiện tại thành giá trị mới
Duyệt	Thuật toán DFS hoặc BFS	Duyệt tất cả các nút trong cây (làm highlight nút hiện tại trong mỗi bước duyệt)
Tìm kiếm	Giá trị tìm kiếm	Tìm kiếm giá trị nút trong cây

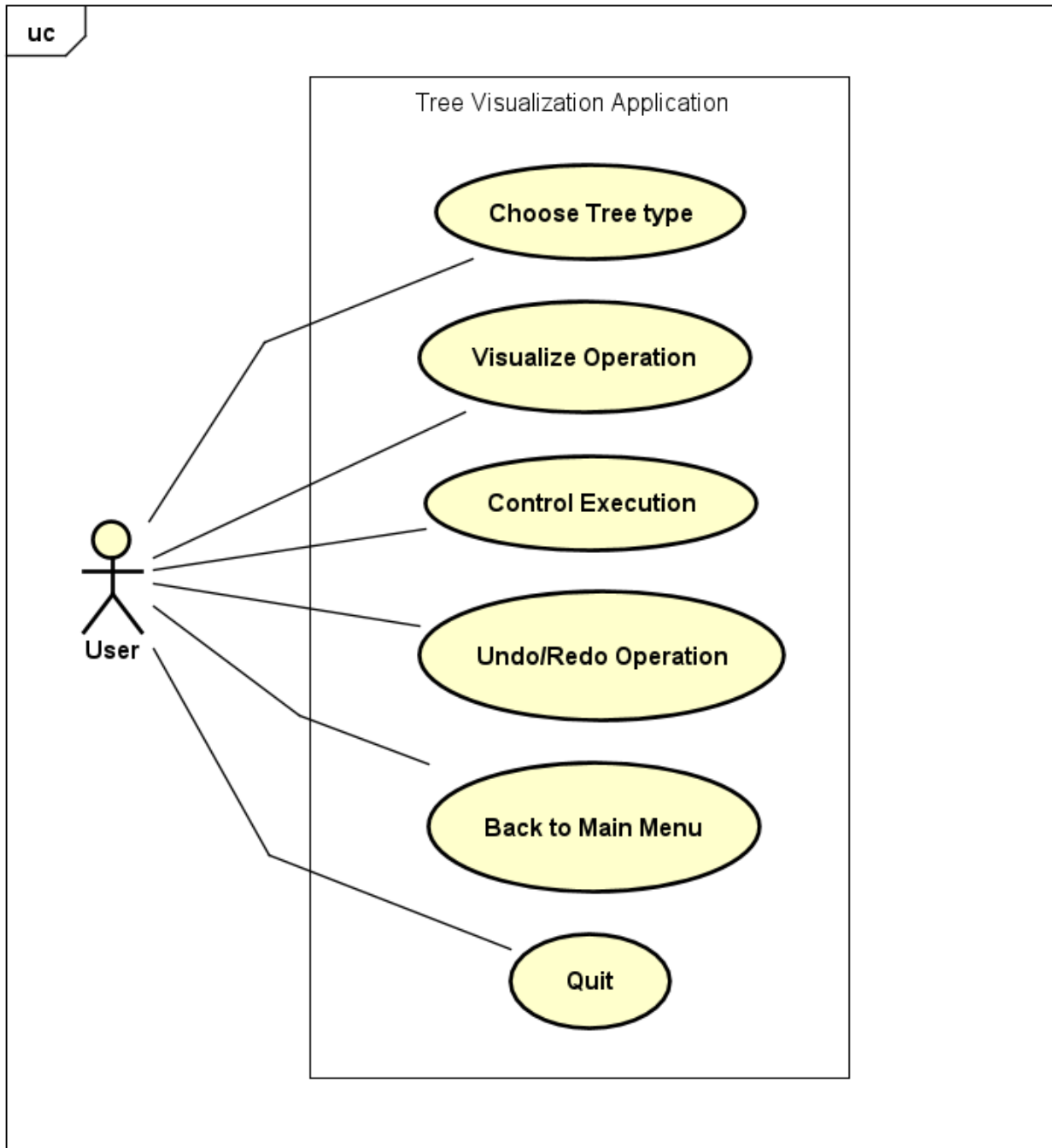
- Khi một thao tác bắt đầu thực thi, trên bảng mã, mã code giả hoặc code thực tế sẽ được hiển thị dòng đang thực thi được làm nổi bật để giúp người dùng theo dõi quá trình. Ở thanh dưới cùng, người dùng có thể xem thanh tiến trình của thao tác đang thực thi và chọn tạm dừng, tiếp tục hoặc đi một bước về phía trước hoặc sau trong quá trình thực thi.
- Người dùng cũng có thể undo hoặc redo các thao tác từ thanh dưới cùng.
- Luôn có nút Quay lại để người dùng quay lại menu chính bất cứ lúc nào.

### 3. Biểu đồ

#### 3.1 General class diagram



#### 3.2 Usecase Diagram



## 4. Mô tả các loại cây và giải thuật

### 4.1 Generic tre:

Định nghĩa: Là tập hợp tất cả các nút được nối với nhau theo quan hệ cha con

Giải thuật:

- Insert:

- Search node cha
  - Insert node con
- Delete:
  - Search node cần xóa
  - Delete
  - Update lại cây
- Update:
  - Search node:
  - Delete node
  - Insert new node
- Traverse: BFS
- Search:
  - Bắt đầu từ root
  - So sánh giá trị với node cần tìm
  - Lặp lại với tất cả node con

#### 4.2 Binary Search Tree (BST)

Định nghĩa: Là cây mà mỗi node không có quá 2 node con, node con trái < node cha, node con phải > node cha

Giải thuật:

- Insert:
  - Search node cha
  - Insert node sang trái/phải phụ thuộc giá trị
- Delete
  - Search node
  - Delete
  - Update lại cây
- Update:
  - Search node:
  - Delete node
  - Insert new node
- Traverse: BFS
- Search:
  - Bắt đầu từ root
  - So sánh giá trị node hiện tại với node cần tìm
  - Đi về bên trái/phải phụ thuộc vào giá trị
  - Lặp lại

### 4.3 Adelson-Velskii Landis (AVL)

Định nghĩa: là BST nhưng có tính chất tự cân bằng dựa trên trọng số

- Insert:
  - Search node cha
  - Insert node sang trái/phải phụ thuộc giá trị
  - Kiểm tra hiệu số cân bằng (balance Factor) dựa trên chiều cao cây con trái - phải
  - Xoay cây nếu lệch
- Balance: xoay cây
  - LL: xoay phải
  - LR: xoay trái, xoay phải
  - RR: xoay trái
  - RL: xoay phải, xoay trái
- Delete: như BST sau đó cân bằng
- Update: Như BST sau đó cân bằng
- Traverse: BFS
- Search: Như BST

### 4.4 Red-Black Tree

Định nghĩa: Cây đỏ đen là một dạng cây tìm kiếm nhị phân tự cân bằng, đảm bảo một số tính chất sau:

- Mọi node trong cây phải là đỏ hoặc đen.
  - Node gốc là đen.
  - Tất cả các lá (NULL) là đen.
  - Mọi node đỏ đều có node con là đen.
  - Tất cả các đường đi từ một nút bất kỳ tới các lá phải đi qua số nút đen bằng nhau.
- 
- Insert: Để chèn một nút vào cây Red-Black, bạn có thể thực hiện các bước sau:
    - 1. Tạo một nút mới với giá trị cần chèn. Màu sắc của nút này sẽ được đặt là RED.
    - 2. Chèn nút này vào cây theo cách tương tự như chèn vào cây nhị phân tìm kiếm.
    - 3. Sau khi chèn, kiểm tra các tính chất của cây Red-Black. Nếu các tính chất này bị vi phạm, thực hiện các hoạt động cần thiết để khôi phục

chúng. Các hoạt động này có thể bao gồm việc xoay cây và đổi màu các nút.

- Delete:
  - Tìm nút cần xóa trong cây theo cách tương tự như tìm kiếm trong cây nhị phân tìm kiếm.
  - Xóa nút này khỏi cây. Nếu nút cần xóa có hai nút con, thì thay thế nút này bằng nút sau cùng trong thứ tự duyệt LNR (nút có giá trị lớn nhất trong cây con trái hoặc nút có giá trị nhỏ nhất trong cây con phải), sau đó xóa nút sau cùng.
  - Sau khi xóa, kiểm tra các tính chất của cây Red-Black. Nếu các tính chất này bị vi phạm, thực hiện các hoạt động cần thiết để khôi phục chúng. Các hoạt động này có thể bao gồm việc xoay cây và đổi màu các nút.
- Update: delete + insert
- Traverse: BFS
- Search: Giống BST

## 5. Phân công công việc

- Nguyễn Lê Sơn - Nhóm trưởng
- Lê Hữu Tài:
  - Research tree data structures and operations
  - AVL
  - Báo cáo + Slide
- Nguyễn Trung Sơn
- Đoàn Bá Sơn