

**Lab 7 - AVL**

**Môn học: Cấu trúc dữ liệu và giải thuật-CSC10004\_22CLC05**

Icon

Description automatically generated with medium confidence

***Giảng viên lý thuyết***

Bùi Tiến Lên

***Giảng viên thực hành***

Trương Tấn Khoa

***Sinh viên thực hiện***

Nguyễn Trương Hoàng Thái 20127625

(2022 – 2023)

**Mục lục**

[Cách chạy chương trình 2](#_Toc142942636)

[Kết quả chạy chương trình 2](#_Toc142942637)

[Bài toán 7.2 2](#_Toc142942638)

[Ý tưởng thực hiện: 2](#_Toc142942639)

[Kết quả chạy chương trình: 3](#_Toc142942640)

[Bài toán 7.3 3](#_Toc142942641)

[Ý tưởng thực hiện: 3](#_Toc142942642)

[Kết quả chạy chương trình: 4](#_Toc142942643)

[Nguồn tham khảo 5](#_Toc142942644)

# Cách chạy chương trình

Chương trình giải quyết các bài toán của lab 7 bao gồm bốn file:

Hai file mã nguồn:

* 7-2.cpp: chương trình giải quyết bài toán 7.2
* 7-3.cpp: chương trình giải quyết bài toán 7.3

Một file dữ liệu (danh sách cây nhị phân cho bài toán 7.2):

* input.txt

File báo cáo:

* Report.pdf

Nhập vào giao diện dòng lệnh theo thứ tự các câu lệnh sau:

1. g++ -o a.exe 7-2.cpp hoặc g++ -o a.exe 7-3.cpp
2. .\a.exe

*Chương trình được biên dịch với trình biên dịch g++ (phiên bản 13.1.0) và các cảnh báo khi biên dịch có thể được bỏ qua.*

# Kết quả chạy chương trình

## Bài toán 7.2

### Ý tưởng thực hiện

Xây dựng các hàm:

* createNode: tạo và trả về một node với số nguyên đầu vào.
* Insert: chèn node chứa giá trị đầu vào vào cây nhị phân tìm kiếm hiện có.
* Height: tính chiều cao của cây nhị phân tìm kiếm (node gốc của độ sâu là 1).
* isAVL: kiểm tra cây nhị phân tìm kiếm truyền vào có phải là cây cân bằng AVL hay không.
* write\_file: ghi kết quả yes (no) vào file Output.txt.
* read\_file: đọc nội dung của file input.txt, thực hiện kiểm tra các cây nhị phân tìm kiếm và ghi kết quả vào file Output.txt.
* main: thực thi hàm read\_file.

Hàm read\_file sử dụng *fgets* để đọc từng dòng của file input.txt. Với mỗi dòng dữ liệu thực hiện:

1. Chuyển các ký tự đọc được sang số nguyên và chèn vào cây nhị phân tìm kiếm.
2. Kiểm tra tính cân bằng của cây nhị phân tìm kiếm. Nếu cây nhị phân tìm kiếm là cân bằng, lưu giá trị “yes” vào mảng kết quả hai chiều. Ngược lại, lưu giá trị “no”.

Cuối cùng, ghi kết quả cuối cùng vào file Output.txt.

### Kết quả chạy chương trình

A screenshot of a computer program

Description automatically generated

1. Sử dụng lệnh dir để kiểm tra các file có trong thư mục hiện tại.
2. Lệnh cat in ra nội dung của file input.txt
3. Sau khi biên dịch và thực thi chương trình, sử dụng lệnh cat để in ra nội dung của file kết quả Output.txt.

## Bài toán 7.3

### Ý tưởng thực hiện

Xây dựng các hàm:

* createNode: tạo và trả về một node với số nguyên đầu vào.
* Height: tính chiều cao của cây nhị phân tìm kiếm (node gốc của độ sâu là 1).
* RightRotation: xoay cây nhị phân phải.
* LeftRotation: xoay cây nhị phân trái.
* MakeRightHeight: cân bằng cây nhị phân tìm kiếm để tạo ra cây nhị phân tìm kiếm cân bằng.
* Insert: chèn node chứa giá trị đầu vào vào cây nhị phân tìm kiếm cân bằng hiện có.
* Remove: xóa node chứa giá trị đầu vào từ cây nhị phân tìm kiếm cân bằng hiện có.
* level: tính toán chiều sâu của node đầu vào trong cây nhị phân đầu vào.
* NLR: tính toán chiều sâu của các node lá.
* main: tạo cây AVL từ mảng số nguyên gồm 6 phần tử, tính toán chiều sâu của các node lá và in kết quả ra màn hình. Tiếp theo đó, xóa đi node chứa khóa 50 trong cây AVL ban đầu và in kết quả ra màn hình sau khi tính toán chiều sâu của cây mới.

### Kết quả chạy chương trình

A black background with white text

Description automatically generated

Cây AVL ban đầu có hình minh họa như sau:

A diagram of a diagram

Description automatically generated

Dựa vào hình minh họa, chúng ta có thể thấy các node lá chứa khóa 10, 25, 50 có cùng độ sâu với nhau.

Sau khi xóa đi node lá chứa khóa 50, cây AVL trở thành:

A diagram of a diagram

Description automatically generated

Sau khi thực hiện phương thức xóa node, các node lá chứa các khóa 10, 25, 40 và các node lá này không có cùng độ sâu.

# Nguồn tham khảo

[1] [CPlusPlus.com](https://cplusplus.com/)

[2] Kiểm tra cây AVL: [tutorialspoint](https://www.tutorialspoint.com/cplusplus-program-to-check-if-a-given-binary-tree-is-an-avl-tree-or-not)

[3] Minh họa cây AVL: [GeeksforGeeks](https://www.geeksforgeeks.org/insertion-in-an-avl-tree/)

[4] Trực quan cây AVL: [cs.usfca.edu](https://www.cs.usfca.edu/~galles/visualization/AVLtree.html)