**TRƯỜNG ĐẠI HỌC FPT**

****

**ASSIGNMENT 2**

**DATA STRUCTURE AND ALGORITHM**

*Người thực hiện*: **Nguyễn Thị Ngọc Thư – MSSV SE171992  
Vũ Hoàng Ân – MSSV SE162132**

Lớp **: SE1856**

Khoá  **: 17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**TRƯỜNG ĐẠI HỌC FPT**

****

**ASSIGNMENT 2**

**DATA STRUCTURE AND ALGORITHM**

*Người thực hiện*: **Nguyễn Thị Ngọc Thư – MSSV SE171992  
Vũ Hoàng Ân – MSSV SE162132**

Lớp **: SE1856**

Khoá  **: 17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**MỤC LỤC**

***Trang***

1. Bài toán

2. Giải quyết bài toán

2.1 Mô tả cấu trúc dữ liệu

2.2 Sơ đồ giải thuật

2.3 Hiện thực

2.4 Kết quả và thảo luận

3. Kết luận

**Tự đánh giá**

**TIÊU ĐỀ ASSIGNMENT 2**

1. **Bài toán**

Với nhu cầu kết nối toàn cầu, hội nhập với thế giới, tiếng anh đã trở thành một phương tiện đắc lực để thiết lập cầu nối. Để việc học tiếng anh hiệu quả hơn ta cần có một từ điển để dễ dàng tra cứu và học hỏi. Mục tiêu hiện tại là tạo ra một ứng dụng từ điển lưu trữ tiếng anh hiệu quả, vấn đề là chúng ta cần lựa chọn một cấu trúc dữ liệu, nó có thể tìm kiếm từ vựng với tốc độ nhanh nhất, dễ chỉnh sửa và thêm xóa sửa nhất và đặt biệt là có khả năng mở rộng trong tương lai. Sau khi cân nhắc chúng tôi tìm thấy AVL hay còn gọi là cây tìm kiếm nhị phân tự cân bằng là phù hợp với điều kiện trên với chỉ số time complexity chỉ O(log(n)) cho mọi tác vụ quan trọng đã được đề cập.

**2. Giải quyết bài toán**

**2.1 Mô tả cấu trúc dữ liệu**

Cấu trúc được sử dụng trong bài toán là AVL Tree hay còn gọi là cây nhị phân tìm kiếm tự cân bằng. Trước khi đề cập đến AVL rõ hơn thì cần làm rõ về Tree và những thành phần trước. Tree là một cấu trúc phân cấp được sử dụng để trình bày và tổ chức dữ liệu để mà dễ dàng điều hướng và tìm kiếm. Nó là tập hợp những Node được nối với nhau bởi các cạnh và có một mối quan hệ giữa các Node. Sau đây là một vài định nghĩa:

* Root: Là node đầu tiên của cây, hay còn gọi là gốc, rễ.
* Internal node: là node với ít nhất một con
* External (leaf) node: là node không có con
* Level of a node: Level của root là một. Nếu một node có level là i thì node con của nó sẽ có level i + 1.
* Height of a tree: Là level cao nhất trong một cây, do đó một node đơn là một cây có level là một.
* Height of a node p: Là chiều cao của cây con với root là p.

Chúng ta quay lại với AVL Tree, một cây bình thường trở thành cây AVL tree khi và chỉ khi sự chênh lệch giữa chiều cao cây bên trái và cây bên phải của một node không thể lớn hơn một, balance factor là một chỉ số để biểu diễn điều đó.

BalanceFactor = height(left-sutree) − height(right-sutree)

Với một cây là AVL tree thì nó sẽ tự cân bằng chính nó khi có một node mới được chèn vào. Do đó, thao tác chèn gây mất thời gian.

*Để làm cho cây tự cân bằng, một cây AVL có thể thực hiện 4 loại kỹ thuật quay sau (1) :*

*Kỹ thuật quay trái :*

Nếu một cây trở nên không cân bằng khi một nút được chèn vào trong cây con bên phải của cây con bên phải thì chúng ta có thể thực hiện kỹ thuật quay trái đơn như sau:

A diagram of a diagram

Description automatically generated with medium confidence

Trong hình minh họa trên, nút A trở nên không cân bằng khi một nút (nút C) được chèn vào cây con bên phải của cây con bên phải của nút A. Chúng ta thực hiện kỹ thuật quay trái để làm A trở thành cây con bên trái của B.

*Kỹ thuật quay phải:*

Cây AVL trở nên không cân bằng nếu một nút được chèn vào cây con bên trái của cây con bên trái. Để cây cân bằng, chúng ta thực hiện kỹ thuật quay phải như sau:

A diagram of a diagram

Description automatically generated

Như hình minh họa, nút không cân bằng sẽ trở thành cây con bên phải của cây con bên trái của nó bằng kỹ thuật quay phải.

*Kỹ thuật quay trái-phải:*

Kỹ thuật quay ghép là khá phức tạp so với hai kỹ thuật quay đơn vừa giới thiệu trên. Để hiểu kỹ thuật quay này nhanh hơn, bạn cần phải ghi chú từng hành động được thực hiện trong khi quay. Một kỹ thuật quay trái-phải là sự kết hợp của kỹ thuật quay trái được theo sau bởi kỹ thuật quay phải.

*Kỹ thuật quay phải-trái:*

Một loại kỹ thuật quay ghép khác là kỹ thuật quay phải-trái. Kỹ thuật này là sự kết hợp của kỹ thuật quay phải được theo sau bởi kỹ thuật quay trái.

*Sau đây là một số ứng dụng của AVL:*

Hầu hết in-memory sets và dictionaries được lưu trữ bằng cách sử dụng cây AVL.

**Các ứng dụng cơ sở dữ liệu, trong đó việc chèn và xóa ít phổ biến hơn nhưng cần phải tra cứu dữ liệu thường xuyên, cũng thường xuyên sử dụng cây AVL.**

**Ngoài các ứng dụng cơ sở dữ liệu, nó còn được sử dụng trong các ứng dụng khác đòi hỏi khả năng tìm kiếm tốt hơn.**

**2.2 Sơ đồ giải thuật**

A diagram of a code

Description automatically generated

A diagram of a function

Description automatically generated

A diagram of a function

Description automatically generated

A diagram of a function

Description automatically generated

A diagram of a system

Description automatically generated A diagram of a computer

Description automatically generated

A diagram of a flowchart

Description automatically generated

A diagram of a flowchart

Description automatically generated

A diagram of a program

Description automatically generated

A flowchart with words

Description automatically generated

A diagram of a flowchart

Description automatically generated

**2.3 Hiện thực**

Dữ liệu:

Đối tượng được kiểm thử là 1 file .txt với 100 dòng dữ liệu, mỗi dòng trình bày một từ tiếng anh và một từ tiếng việt với ý nghĩa tương ứng, được ngăn cách nhau bởi dấu “,”.

Công nghệ sử dụng:

|  |  |
| --- | --- |
| Ngôn ngữ | Java 1.8 |
| Thư viện | Collection, Util |
| Môi trường | Netbean IDE 8.2 |

1. Cấu trúc dự án

A screenshot of a computer

Description automatically generated

1. Node.java

A screenshot of a computer code

Description automatically generated

1. AVLtree.java

* Balance factor:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

- Insert :

A screenshot of a computer code

Description automatically generated

* Delete

A computer code with black text

Description automatically generated

A computer screen shot of a program code

Description automatically generated

A screenshot of a computer code

Description automatically generated

* Search

A screen shot of a computer code

Description automatically generated

* FindPath

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

**2.4 Kết quả và thảo luận**

Kết quả đạt được:

|  |  |  |
| --- | --- | --- |
| Giải thuật | Time Complexity | Tỉ lệ hoàn thành |
| Insert | O(logn) | 100% |
| Delete | O(logn) | 100% |
| Search | O(logn) | 100% |
| Build simple balanced tree | O(logn) | 100% |
| FindPath | O(logn) | 100% |

Kết quả đạt được và thảo luận:

Về mặt lý thuyết: để giải được bài toán, yêu cầu phải tìm hiểu được những phương pháp, hướng đi và cách hoạt động của Simple balanced tree. Nghiên cứu về cách hoạt động của AVL tree, ứng dụng thành công những thuật toán vào việc giải quyết vấn đề cụ thể.

Về mặt thực hiện: Giải quyết thành công việc thêm một từ mới bằng việc áp dụng AVL tree, sau mỗi lần cập nhật một giá trị cây sẽ tự động cân bằng và cập nhật lại chiều cao. Giải quyết thành công xóa một từ, sau khi xóa cây tiếp tục cập nhật lại các node thông qua phương pháp DeletebyMerge và DeletebyCopy. Hoàn thiện chức năng search, có thể trả lại thông tin tương ứng với tham số được truyền vào. Hoàn thành việc phân thích và triển khai việc in ra đường đi từ một node đến một node khác một cách hiệu quả.

**3. Kết luận**

Bài học:

1. Có cơ hội được triển khai ứng dụng cấu trúc dữ liệu cây để giải quyết một bài toán cụ thể.
2. Cải thiện cách phân tích thuật toán nằm đáp ứng việc giải quyết mỗi vấn đề nhỏ trong bài toán lớn.
3. Thông qua việc nghiên cứu và ứng dụng về tree, cũng mang lại cơ hội để tiếp cận, so sánh và mở rộng sang những cấu trúc dữ liệu khác.
4. Cải thiện khả năng tự đánh giá, so sánh các cấu trúc dữ liệu, các cách giải thuật khác nhau để tìm ra cách tối ưu và phù hợp nhất cho từng vấn đề.

Hướng phát triển:

Bản thân: Tự build cho bản thân 1 app có thể lưu trữ từ vựng mở rộng từ dự án này nhằm mục đích sử dụng cho việc học tiếng Anh

Dự án: Mở rộng ra để làm 1 application áp dụng giải thuật cây cân bằng để tăng hiệu xuất truy xuất giữa server và client

**TỰ ĐÁNH GIÁ – ĐỀ TÀI NHÓM**

|  |  |  |
| --- | --- | --- |
| **Nội dung** | **Điểm** | **Ghi chú** |
| Giới thiệu về bài toán (0.25 đ) | 0.25 |  |
| Mô tả cấu trúc dữ liệu (1.25 đ) | 1.25 |  |
| Sơ đồ giải thuật (1.5 đ) | 1.5 |  |
| Hiện thực (5 đ) | 5 |  |
| Kết quả và thảo luận (0.5 đ) | 0.5 |  |
| Điểm nhóm (0.75 đ) | 0.75 |  |
| Các điều rút ra cho bản thân (0.25 đ) | 0.25 |  |
| Báo cáo đúng theo mẫu (0.5 đ) | 0 |  |
| Tổng điểm | **9.5** |  |

**TỰ ĐÁNH GIÁ- ĐỀ TÀI 1 THÀNH VIÊN**

|  |  |  |
| --- | --- | --- |
| **Nội dung** | **Điểm** | **Ghi chú** |
| Giới thiệu về bài toán (0.25 đ) | 0.25 |  |
| Mô tả cấu trúc dữ liệu (1.25 đ) | 1.25 |  |
| Sơ đồ giải thuật (1.5 đ) | 1.5 |  |
| Hiện thực (5.75 đ) | 5.75 |  |
| Kết quả và thảo luận (0.5 đ) | 0.5 |  |
| Các điều rút ra cho bản thân (0.25 đ) | 0.25 |  |
| Báo cáo đúng theo mẫu (0.5 đ) | 0 |  |
| Tổng điểm | **9.5** |  |

**Chú ý: Nếu không chạy được chương trình, chỉ tính tối đa 4 điểm (Giới thiệu bài toán: 0.25, Mô tả cấu trúc dữ liệu: 1.25, Sơ đồ giải thuật: 1.5, Hiện thực: 0.5 và Báo cáo: 0.5)**

**TÀI LIỆU THAM KHẢO**

1. **https://quantrimang.com/cong-nghe/cay-avl-trong-cau-truc-du-lieu-va-giai-thuat-156515**