Data Structures and Algorithms

HUS HKI, 23 - 24

Assignment 7

Lecturer: Nguyễn Thị Hồng Minh
Trần Bá Tuấn - Đặng Trung Du

§ Searching and Binary Search Trees §

Phần 1: THÔNG TIN CHUNG

(1) Mục tiêu

- Nắm được các khái niệm, kiến thức liên quan đến thuật toán tìm kiếm tuần tự, tìm kiếm nhị phân và cài đặt được các thuật toán.
- Nắm được khái niệm cây tìm kiếm nhị phân, các tính chất và các thao tác tìm kiếm/thêm/xóa phần tử.

(2) Tài liệu đọc thêm

- Binary Search Tree - Binary Search Tree
- Binary Search Tree | Set 1 (Search and Insertion) - Binary Search Tree | Set 1 (Search and Insertion)
- Binary Search Tree | Set 2 (Delete) - Binary Search Tree | Set 2 (Delete)

(3) Quy cách nộp bài

- Mỗi sinh viên hoàn thành bài tập trong package có tên Hw7_<idSinhvien>_<Hovaten>
- Trong đó, <idSinhvien> là mã sinh viên.
- Sinh viên nộp bài làm trên tài khoản của mình bao gồm:
  1. File nén .zip của thư mục chứa package (Hw7_<idSinhvien>_<Hovaten>.zip)
  2. Tất cả các file nguồn *.java.
- Khi hết hạn nộp bài, các bài làm sẽ công bố để thực hiện chấm chéo bài tập.
- Sinh viên không nộp bài sẽ nhận điểm 0 bài tập tuần.
- Sinh viên CÓ GIAN LẬN trong nộp bài tập sẽ bị ĐÌNH CHỈ môn học (điểm 0 cho tất cả các điểm thành phần).

Phần 2: Thực hành

(1) Bài tập

CHÚ Ý

1. Lựa chọn các gói bài tập để thực hiện:
   - Combo 1: Bài 1, Bài 2 và tự chọn 4 bài trong phần Bài tập thêm (Cơ bản) được 80đ.

- Combo 2: Bài 1, Bài 2, Bài 3 và tự chọn 2 bài trong phần Bài tập thêm (Nâng cao 1) được 90đ.
- Combo 3: Bài 2, Bài 3, Bài 4 và tự chọn 2 bài trong phần Bài tập thêm (Nâng cao 2) được 100đ.

2. Trong bài nộp có file .doc hoặc .txt thuyết minh về gói bài tập thực hiện và những nội dung cần giải thích về bài làm: thuật toán được chọn, tài liệu tham khảo, định dạng input, yêu cầu hệ thống...

Bài tập 1. *Thực hiện lại các thuật toán tìm kiếm trên các danh sách cài đặt bằng mảng, cấu trúc liên kết có sắp xếp và không sắp xếp.*
Sử dụng lại giao diện và các kiểu dữ liệu đã cài đặt trong ***Homework 3****.*

Bài tập 2. *Xây dựng* ***Cây nhị phân tìm kiếm*** *với các khóa nguyên, với các phương thức*

- ***FindMin()****: Tìm phần tử nhỏ nhất trên cây.*
- ***Search(x, T)****: Tìm phần tử có giá trị x trên cây T.*
- ***Insert(x, T)****: Thêm phần tử x vào cây T.*
- ***Delete(x, T)****: Xóa phần tử x từ cây T.*

Sử dụng kiểu dữ liệu ***Binary Tree*** *cài đặt bằng cấu trúc liên kết đã thực hiện trong* ***Homework 5****.*
So sánh thời gian thực hiện các thuật toán tìm kiếm tuần tự, tìm kiếm nhị phân và tìm kiếm trên cây tìm kiếm với tập các số nguyên có kích thước lớn ($10^6, 10^7, 10^8$).

Bài tập 3. *Xây dựng cây cân bằng trên cơ sở mở rộng cây nhị phân tìm kiếm ở bài 2 với các hàm xoay (Rotation).*

Bài tập 4. *Triển khai một ứng dụng của cây nhị phân tìm kiếm.*
Gợi ý:

- *Line segment intersection*
- *Range search in 2-d trees*
- *Nearest neighbor search in a 2d tree*
- *Rectangle intersection*

(2) Bài tập thêm

Luyện tập 1. *Kiểm tra một dãy có là dãy duyệt giữa của cây tìm kiếm nhị phân - Inorder Traversal and BST*

Luyện tập 2. *Kiểm tra xem một cây có là cây tìm kiếm nhị phân - Check for BST*

- *Nguồn: A program to check if a binary tree is BST or not*

Luyện tập 3. *Tìm Median của Cây tìm kiếm nhị phân - Median of BST*

Luyện tập 4. *Đếm các giá trị của cây trong khoảng cho trước - Count BST nodes that lie in a given range*

- *Thực hiện duyệt cây và kiểm tra nút đang xét có nằm trong đoạn cần đếm.*

Luyện tập 5. *Phần tử nhỏ nhất thứ k trong cây tìm kiếm nhị phân - k-th smallest element in BST*

- *Duyệt giữa cây tìm kiếm nhị phân và in ra phần tử thứ k của dãy kết quả.*

Luyện tập 6. *Phần tử lớn thứ k trong cây tìm kiếm nhị phân - Kth largest element in BST*

- *Duyệt giữa cây tìm kiếm nhị phân và in ra phần tử thứ k từ phải sang của dãy kết quả.*

Luyện tập 7. *Phần tử bé nhất trên cây tìm kiếm nhị phân - Minimum element in BST*

Luyện tập 8. *Tìm phần tử liền sau của một nốt trên cây tìm kiếm nhị phân - Inorder Successor in BST*

- *Nguồn: Inorder Successor in Binary Search Tree*

Luyện tập 9. *Tìm phần tử liền trước và liền sau của một phần tử trên cây tìm kiếm nhị phân - Predecessor and Successor*

- *Nguồn: Inorder predecessor and successor for a given key in BST*
- ***Gợi ý:** Đơn giản nhất là duyệt giữa, thực hiện tìm kiếm số cần tìm và in ra hai số liền trước và liền sau.*

Luyện tập 10. *Tìm giá trị phần tử lớn nhất trên cây tìm kiếm nhị phân mà không vượt quá N - Closest Neighbor in BST*

Luyện tập 11. *Tìm số bé nhất trên cây tìm kiếm nhị phân mà có giá trị lớn hơn hoặc bằng một giá trị cho trước - Ceil in BST*

Luyện tập 12. *Biến đổi một cây sang cây tìm kiếm nhị phân -Binary Tree to BST*

Luyện tập 13. *Tạo cây tìm kiếm nhị phân từ các phép duyệt sau -Construct BST from Postorder*

- *Nguồn: Construct a Binary Search Tree from given postorder*