TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Báo cáo Combinatorial Optimization**

*Người hướng dẫn*: **CÔ THÂN LÊ**

*Người thực hiện*: **PHẠM HOÀNG KHÔI NGUYÊN - 51703143**

Lớp: 17050301

Khoá  **: 21**

LỜI CẢM ƠN

Đầu tiên tôi xin gửi lời cảm ơn đến các thầy cô trong Khoa Công Nghệ Thông Tin vì đã mang dạy cho chúng tôi những kiến thức hay và có ích ,với sự nhiệt tình giảng dạy của thầy cô đã mang chúng tôi đến một vùng trời kiến thức.Đặc biệt tôi xin gửi lời cảm ơn trân trọng nhất đến thầy Lê vì thầy đã mang đến rất nhiều kiến thức cơ bản cũng như nâng cao của môn cấu trúc dữ liệu,với những kiến thức đó tôi đã có thể hoàn thành bài tập lớn lần này và cũng như có được những kiến thức này để giúp ích cho tương lai tôi khi đi ra xã hội.

Sau cùng,tôi xin kính chúc quý Thầy Cô trong Khoa Công Nghệ Thông Tin và thầy Lê dồi dào sức khỏe và niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

TIỂU LUẬN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm tiểu luận của riêng tôi và được sự hướng dẫn của Thầy Thắng;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 17 tháng 10 năm 2018

Tác giả

(ký tên và ghi rõ họ tên)

Nguyên

Phạm Hoàng Khôi Nguyên

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

Tóm tắt

Phân tích và thiết kế chương trình phù hợp với 3 bài toán:tìm đường đi ngắn nhất,tìm cây khung ngắn nhất ,tìm đường đi lớn nhất

Mục Lục

[LỜI CẢM ƠN 1](#_Toc6520854)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN 3](#_Toc6520855)

[Tóm tắt 4](#_Toc6520856)

[Mục Lục 5](#_Toc6520857)

1. Cây BST
   1. Hàm tạo cây
      1. Hàm tạo cây rỗng

createTree():không truyền bất kì tham số nào vào ,khi tạo cây cả cây cũ sẽ bị xóa.

* + 1. Hàm tạo cây theo danh sách

createTree(ArrayList<SV> sv):truyền vào danh sách các sinh viên, sẽ thêm các sinh viên trong danh sách vào cây cho tới cuối, xóa cây cũ.

* + 1. Hàm tạo cây ngẫu nhiên

createTreeRNG(int size):truyền vào 1 số sẽ tạo ra cây có kích thước bằng với số truyền vào, xóa cây cũ.

* + 1. Hàm tạo cây lệch trái

createTreeleft(int size):tạo cây RNG như bình thường ,sau đó sau xoay root sang trái cho tới khi kiểm tra checkbal(root) lệch trái.

createTreeleft(ArrayList<SV> sv):tạo cây theo danh sách ,sau đó sau xoay root sang trái cho tới khi kiểm tra checkbal(root) lệch trái.

* + 1. Hàm tạo cây lệch phải

createTreeright(int size):tạo cây RNG như bình thường ,sau đó sau xoay root sang phải cho tới khi kiểm tra checkbal(root) lệch phải.

createTreeright(ArrayList<SV> sv):tạo cây theo danh sách ,sau đó sau xoay root sang phải cho tới khi kiểm tra checkbal(root) lệch phải.

* 1. Hàm duyệt cây

L=Left ,N=Node,R=Right

Duyệt theo thứ tự của các hàm LNR, LRN, NLR, RNL, NRL, RLN. In ra danh sách theo thứ tự MSSV, họ tên, ngày sinh, điểm trung bình, số tín chỉ tích lũy theo thứ tự của hàm được gọi

* 1. Hàm tìm kiếm (chỉ trả vể node đầu tiên tìm được)
     1. Tìm theo MSSV

searchMSSV(int MSSV):Dò trên cả danh sách để tìm MSSV và trả về node đó.Nếu node không tồn tại trả về null.

* + 1. Tìm theo tên

searchTen(String Ten):Dò trên cả danh sách để tìm tên và trả về node đó.Nếu node không tồn tại trả về null.

* + 1. Tìm theo ngày sinh

searchNgaySinh(String ngaysinh):Dò trên cả danh sách để tìm ngày sinh và trả về node đó.Nếu node không tồn tại trả về null.

* + 1. Tìm theo điểm trung bình

searchDTB(int dtb):Dò trên cả danh sách để tìm điểm trung bình và trả về node đó.Nếu node không tồn tại trả về null.

* + 1. Tìm theo tín chỉ tích lũy

searchTCTL(double tctl):Dò trên cả danh sách để tìm điểm tích lũy và trả về node đó.Nếu node không tồn tại trả về null.

* + 1. Tìm phần tử lớn nhất

Max():Đi về phía bên trái từ node được chọn khi đụng null thì trả về node trước đó.

* + 1. Tìm phần tử nhỏ nhất

Min():Đi về phía bên phải từ node được chọn khi đụng null thì trả về node trước đó

* 1. Hàm thêm phần tử vào cây
     1. Thêm từng node

add(SV sv):Thêm 1 sinh viên vào cây

* + 1. Thêm cả danh sách

add(ArrayList<SV> sv):Thêm 1 danh sách các sinh viên vào cây

* 1. Hàm xóa phần tử trong cây
     1. Xóa từng node

remove(int mssv):Xóa 1 sinh viên trong cây

* + 1. Xóa cả sanh sách

remove(ArrayList<Integer> mssv):Xóa danh sách các sinh viên trong cây

* 1. Hàm tìm phần tử liền trước (predecessor(Node sv))

Trường hợp 1:Node có nhánh cây bên trái

Trả về giá trí lớn nhất của nhánh cây trái hoặc trả về nhánh trái

Trường hợp 2:Node không có nhánh cây bên trái

Tìm kiếm parent phù hợp điều kiện sau để trả về:

-phải nằm trên cùng nhánh với node cần tìm kiếm

-phải là parent cuối cùng có con là node cần tìm kiếm hoặc là node parent thứ n của node cần tìm kiếm nằm ở bên trái của parent đó.

* 1. Hàm tìm phần tử liền sau(Successor(Node SV)

Hàm này có cách thực hiện ngược lại so với hàm trên

* 1. Hàm cập nhật thông tin trên của sinh viên(update(SV sv))

Hàm này tìm kiếm và thay thế thông tin sinh viên có trong danh sách theo MSSV nếu sinh viên không có trong danh sách thì sẽ thêm sinh viên đó vào danh sách.Hàm nhận vào dữ liệu sinh viên .

1. Cây AVL(có phần lớn các hàm giống hàm của cây BST)
   1. Hàm xoay trái ( rotateRight(Node y))

Xoay node được chọn về bên trái đồng thời cập nhật lại độ cao và parent của cây

* 1. Hàm xoay phải ( rotateLeft(Node x))

Xoay node được chọn về bên phảiđồng thời cập nhật lại độ cao và parent của cây

* 1. Kiểm tra cân bằng của cây ( checkbal(Node x))

Kiểm tra cân bằng bằng cách lấy độ cao của nhánh trái trừ cho nhánh phải nếu lớn hơn 1 thì lệch trái nếu nhỏ hơn -1 thì lệch phải

* 1. Cân bằng cây ( rebalance(Node x))

Dùng phương pháp xoay trái và phải của các node và kiểm trả cân bằng để cân bằng lại cây