**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN HỌC**

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**Giảng viên hướng dẫn: TS. Hoàng Văn Thông**

**Sinh viên thực hiện: Phạm Duy Lợi**

**Mã SV: 222601120**

**Lớp: CNTT – VA1**

**Hà Nội, tháng 11 năm 2023**

Mục lục

[**BÀI 1 – BÀI SỐ 35 TRONG DANH SÁCH BÀI TẬP** 3](#_Toc151759395)

[**I.** **Đề bài** 3](#_Toc151759396)

[**II.** **Phân tích bài toán** 4](#_Toc151759397)

[**III.** **Cài đặt các lớp và hàm main bằng C++** 5](#_Toc151759398)

[**IV.** **Phân tích thời gian chạy của các phương thức có trong lớp:** 6](#_Toc151759399)

[**BÀI 2 – BÀI SỐ 7 TRONG DANH SÁCH BÀI TẬP** 8](#_Toc151759400)

[**I.** **Đề bài** 8](#_Toc151759401)

[**II.** **Phân tích bài toán** 9](#_Toc151759402)

[**III.** **Cài đặt các lớp và hàm main vào C++** 10](#_Toc151759403)

[**IV.** **Phân tích thời gian chạy của các phương thức có trong lớp** 14](#_Toc151759404)

# **BÀI 1 – BÀI SỐ 35 TRONG DANH SÁCH BÀI TẬP**

1. **Đề bài**

Cài đặt lớp danh sách liên kết kép mẫu, bổ sung phương thức sắp xếp các phần tử theo thứ tự tăng dần hoặc dần tùy thuộc vào tham số.

Ứng dụng giải bài toán sau:

Một buổi họp mặt đại gia đình nhân dịp cụ già Ted tròn 100 tuổi, người ta muốn sắp xếp con cháu của cụ theo thứ tự từ tuổi cao xuống thấp. Giả sử ta có thông tin về giấy khai sinh của từng người đó. Mỗi giấy khai sinh chỉ viết ba thông tin đơn giản gồm: Tên người cha, Tên người con, Tuổi của người cha lúc sinh con.

Hãy giúp đại gia đình trên tính ra tuổi của từng người con cháu cụ Ted và viết ra danh sách theo thứ tự từ tuổi cao xuống thấp.

Đầu vào:

Dòng đầu ghi số bộ test (không quá 100). Với mỗi bộ test:

- Dòng đầu tiên ghi số X (0<X<100) là số người con cháu cần sắp xếp.

- Tiếp theo là X dòng, mỗi dòng ghi thông tin về một giấy khai sinh của từng người (thứ tự ngẫu nhiên) gồm 3 thành phần, mỗi thành phần cách nhau một khoảng trống:

o Tên người cha: không quá 20 ký tự và không chứa khoảng trống

o Tên người con: không quá 20 ký tự và không chứa khoảng trống

o Tuổi của người cha khi sinh con: 1 số nguyên dương, không quá 100.

Đầu ra:

- Với mỗi bộ test, in ra màn hình thứ tự bộ test (xem thêm trong bộ test ví dụ), sau đó lần lượt là từng người trong danh sách tuổi từ cao xuống thấp (không tính cụ Ted). Mỗi người viết ra hai thông tin: tên, một khoảng trống rồi đến tuổi của người đó.

- Nếu hai người có cùng tuổi thì xếp theo thứ tự từ điển.

Ví dụ:

Input:

2

1

Ted Bill 25

4

Ray James 40

James Beelzebub 17

Ray Mark 75

Ted Ray 20

Output:

DATASET 1

Bill 75

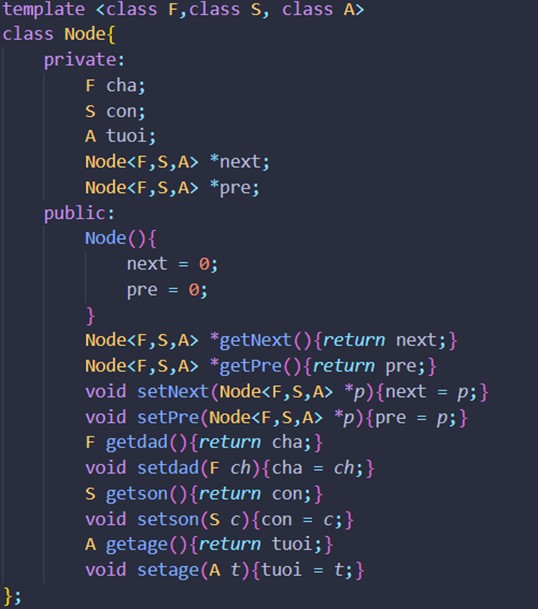
DATASET 2

Ray 80

James 40

Beelzebub 23

Mark 5

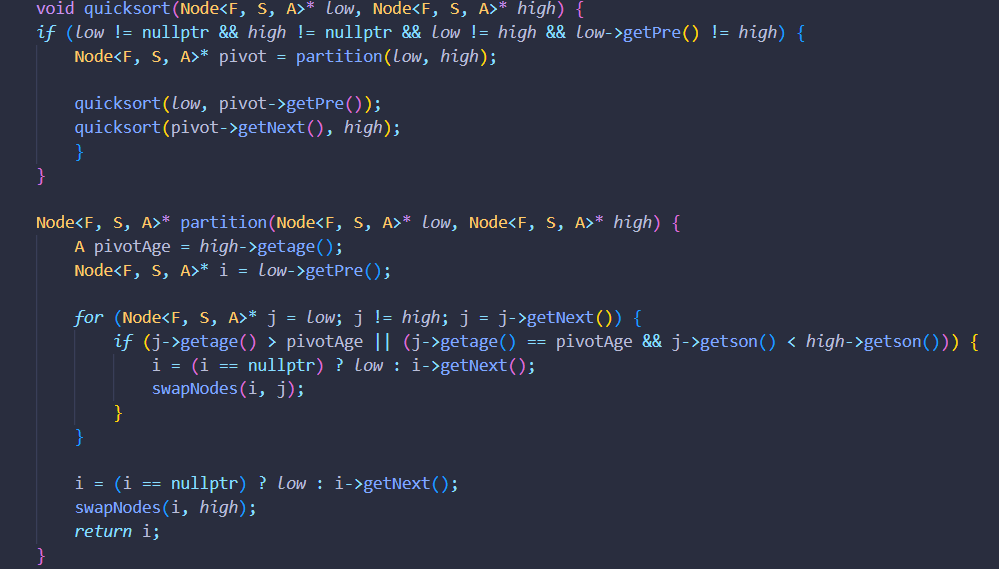
1. **Phân tích bài toán**
   * Bài toán yêu cầu cài đặt dánh sách liên kết kép, bổ sung phương thức sắp xếp, sau đó ứng dụng giải bài toán tìm tuổi của các con cháu của Ted và sắp xếp theo thứ tự giảm dần, nếu cùng tuổi thì sắp xếp tên theo thứ tự từ điển
   * Với danh sách liên kết kép, mỗi phần tử sẽ liên kết với phần tử đứng trước và sau nó trong danh sách. Mỗi phần tử trong danh sách (node) gồm biến lưu dữ liệu và 2 con trỏ liên kết tới phần tử trước và sau nó.
   * Các phương thức xây dựng:
     + Tạo một danh sách các con cháu của Ted
     + Thêm con cháu vào danh sách
     + Sắp xếp danh sách theo sao cho tên người cha luôn xuất hiện trước người con
     + Tính tuổi của các con cháu của Ted
     + Sắp xếp các con cháu theo tuổi và theo thứ tự từ điển
     + Hiển thị danh sách ra màn hình
   * Các lớp xây dựng trong bài:
     + Node
     + Doublelist
   * Phương thức trong Node:
     + GetNext: trả về địa chỉ Node tiếp theo
     + GetPre: trả về địa chỉ Node trước đó
     + SetNext: gắn giá trị cho thuộc tính Next
     + SetPre: gắn giá trị cho thuộc tính Pre
     + Setdad(F ch): gắn giá trị tên cha cho thuộc tính F
     + Setson(S c): gắn giá trị tên con cho thuộc tính S
     + Setage(A t): gắn giá trị tuổi cha khi sinh con cho thuộc tính A
     + Getdad: trả về địa chỉ của tên cha
     + Getson: trả về địa chỉ của tên con
     + Getage: trả về tuổi
   * Phương thức trong Doublelist:
     + Doublelist: khởi tạo danh sách liên kết kép rỗng ban đầu
     + Node pushback: thêm Node vào cuối danh sách
     + Node pushfront: thêm Node vào đầu danh sách
     + Void remove: xóa Node
     + Void insert: chèn Node
     + Void swapnode: hoán đổi giá trị 2 Node
     + Void push: Nhập danh sách con cháu của Ted
     + Void sortdad: Sắp xếp danh sách sao cho tên cha luôn xuất hiện trước tên con
     + Void calcul: tính tuổi các con cháu
     + Void sort: sắp xếp theo tuổi và theo thứ tự từ điển
     + Void printf: in ra màn hình
2. **Cài đặt các lớp và hàm main bằng C++**
   * Đầu tiên xây dựng class Node: 
   * Xây dựng class Doublelist và import Node đã cài đặt vào: A screen shot of a computer program

     Description automatically generated
   * Hàm main: A computer screen shot of code

     Description automatically generated
3. **Phân tích thời gian chạy của các phương thức có trong lớp:** 
   * Các phương thức trong class Node đều có thời gian chạy O(1)
   * Push: Thời gian chạy O(n)A computer code with text

     Description automatically generated
   * Sortdad: Thời gian chạy O(n^2)A screen shot of a computer code

     Description automatically generated
   * Calcul: Thời gian chạy O(n^2)A computer screen shot of code

     Description automatically generated
   * Sort: sử dụng thuật toán QuickSort, thời gian chạy O(nlogn)
   * Printf: Thời gian chạy O(n)A computer code with text

     Description automatically generated with medium confidence

# **BÀI 2 – BÀI SỐ 7 TRONG DANH SÁCH BÀI TẬP**

1. **Đề bài**

1. Xây dựng lớp biểu diễn một bảng băm sử dụng một trong những hàm băm đã học

o Sử dụng lớp bảng băm để xây dựng lớp biểu diễn từ điển Anh Việt. Mỗi phần tử của từ điển là một cặp (tiếng anh, nghĩa tiếng việt). Với các phương thức:

a. Nạp từ điển từ file vào bảng băm

b. Tìm kiếm bằng phương pháp tìm kiếm trên bảng băm

c. Sửa đổi các từ

d. Thêm từ mới

e. Xóa từ

f. Lưu từ điển vào file

2. Xây dựng chương trình có các chức năng

a. Nạp từ điển từ file (có thể nạp tự động khi chạy chương trình)

b. Tra từ điển (nhập vào một từ, hiển thị nghĩa của nó nếu có trong từ điển)

c. Sửa đổi từ

d. Thêm từ mới

e. Xóa từ

f. Lưu từ điển vào file

1. **Phân tích bài toán**
   * Bài toán yêu cầu xây dựng lớp biểu diễn bảng băm và sử dụng hàm băm đã học để xây dựng chương trình từ điển có thể thêm từ, tra từ, sửa đổi từ,xóa từ, lưu vào file.
   * Bảng băm là 1 cấu trúc mà khi người dùng thực hiện truy xuất một phần từ qua khóa thì nó sẽ được ánh xạ vào thông qua hàm băm.
   * Các phương thức xây dựng:
     + Tạo bảng băm chứa từ điển anh việt
     + Đọc file có sẵn
     + Lưu vào file
     + Tra từ điển
     + Thêm từ
     + Xóa từ
     + Hiển thị ra màn hình từ điển
     + Hàm băm (sử dụng phương pháp djb2) -Daniel J.Bernstein
   * Các lớp xây dựng trong bài:
     + Node
     + SingleList
     + Hashtable
     + Dictionary
   * Các phương thức trong Node:
     + Getkey: trả lại địa chỉ của từ tiếng anh
     + Setkey: gắn giá trị từ tiếng anh vào Keys
     + getNext: trả lại địa chỉ Node kế tiếp
     + setnext: gắn địa chỉ cho thuộc tính Next
     + getElem: trả lại địa chỉ của nghĩa tiếng việt
     + setElem: gắn giá trị của từ tiếng việt vào Elem
   * Các phương thức trong SingleList:
     + Node getHead: trả về địa chỉ của Node đầu tiên
     + SingleList : khởi tạo danh sách liên kết đơn rỗng ban đầu
     + Long size: trả về số lượng các phần từ trong danh sách
     + Int isEmpty: kiểm tra xem danh sách có rỗng không
     + Node pushback: thêm Node vào cuối danh sách
     + Void replace: thay thế Node cũ bằng Node mới
     + Void remove: xóa Node
     + Node getNode: trả về giá trị của Node cần tìm
   * Các phương thức trong Hashtable:
     + Hashtable: tạo bảng băm và tạo danh sách liên kết để lưu trữ các phần tử trong bảng băm
     + ~hashtable: giải phóng bộ nhớ, xóa các danh sách liên kết
     + Add: thêm phần tử vào bảng băm, nếu tồn tại thì trả về Null còn không thì trả về phần tử vừa thêm
     + Remove: xóa phần từ khỏi bảng băm
     + Find: Tìm kiếm phần tử trong bảng băm
     + Count: đếm số lượng phần tử trong bảng băm
     + Getsize: trả về kích thước của bảng băm
     + Gettable: trả về bảng băm chứa các danh sách liên kết
   * Các phương thức trong Dictionary:
     + Dictionary: khởi tạo từ điển
     + LoadFromFile: đọc file và thêm dữ liệu vào từ điển
     + SaveToFile: lưu dữ liệu vào File
     + Lookup: tìm kiếm nghĩa trong từ điển
     + Modify: Sửa đổi nghĩa của từ đã có trong từ điển
     + Remove: xóa từ khỏi từ điển
     + Display: hiển thị danh sách từ điển ra màn hình
     + Hash: hàm băm
2. **Cài đặt các lớp và hàm main vào C++**
   * Đầu tiên xây dựng class Node: A computer screen shot of code

     Description automatically generated
   * Xây dựng class SingleList và import Node đã cài đặt vào: A screen shot of a computer code

     Description automatically generated
   * Xây dựng class Hashtable vào import SingleList vào: A computer screen shot of code

     Description automatically generated
   * Xây dựng class Dictionary và import Hashtable vào: A screenshot of a computer program

     Description automatically generated
   * Hàm Main: A screen shot of a computer program

     Description automatically generatedA screen shot of a computer code

     Description automatically generated
3. **Phân tích thời gian chạy của các phương thức có trong lớp**
   * Các phương thức trong Class Node và Class SingleList có thời gian chạy O(1)
   * Class Hashtable:
     + Hastable: thời gian chạy O(n) A computer screen with text

       Description automatically generated
     + Add: thời gian chay O(1) A computer code with text

       Description automatically generated with medium confidence
     + Remove and Find: thời gian chạy O(1) A computer code with many colorful text

       Description automatically generated with medium confidence
     + Count: thời gian chạy O(m+n) A computer screen shot of a number

       Description automatically generated
   * Class Dictionary:
     + Dictionary: thời gian chạy O(n)
     + LoadFromFile: thời gian chạy O(n) A screen shot of a computer code

       Description automatically generated
     + SavetoFile: thời gian chạy O(m+n) A computer screen shot of code

       Description automatically generated
     + Lookup, Modify, Addword, RemoveWord: thời gian chạy: O(1) đến O(n) tùy thuộc vào cách phân phối trong hashtableA computer screen shot of text

       Description automatically generated
     + Display: thời gian chạy O(m+n) A computer screen shot of code

       Description automatically generated