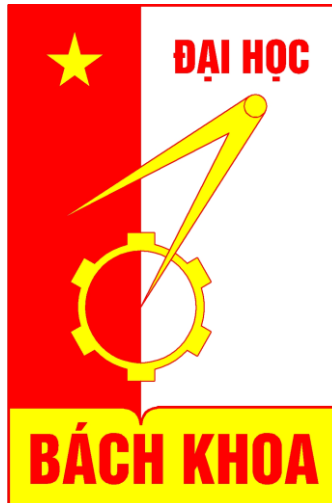


**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI VIỆN CÔNG NGHỆ  
THÔNG TIN & TRUYỀN THÔNG**

-----□-----



**BÁO CÁO BÀI TẬP  
MÔN LẬP TRÌNH C NÂNG CAO**

**Giáo viên hướng dẫn: Thầy Nguyễn Bá Ngọc**

Thành viên nhóm

Lê Văn Duẩn	20194505
Nguyễn Hải Dương	20194530
Đình Việt Long	20194608

**Source code:** [https://github.com/LeVDuan/Phone\\_book\\_app.git](https://github.com/LeVDuan/Phone_book_app.git)

# MỤC LỤC

1. Phân tích bài toán:.....	3
2. Cơ sở lí thuyết:.....	3
2.1. Cây đồ đen .....	3
2.2. GTK: .....	6
2.3. Glade:.....	7
2.4. Thư viện Cgen: .....	7
3. Thiết kế phần mềm:.....	8
3.1. Chức năng .....	8
3.2. Tổ chức bộ nhớ lưu trữ: .....	8
3.3. Quy trình thực hiện:.....	9
3.4. Thiết kế giao diện và vận hành: .....	10
4. Phân công và kế hoạch công việc:.....	12
5. Tài liệu tham khảo:.....	12

# **1. Phân tích bài toán:**

Danh bạ điện thoại là một ứng dụng phần mềm rất quen thuộc đối với người dùng điện thoại nói chung, và việc tra cứu số điện thoại và thông tin cá nhân cơ bản trong danh bạ là không thể thiếu trong cuộc sống sinh hoạt hàng ngày.

Qua bài tập lớn lần này, chúng em muốn giới thiệu một phần mềm quản lý danh bạ điện thoại bằng ngôn ngữ lập trình C với giao diện đồ họa gtk3 và Glade. Phần mềm này sẽ đạt được các yêu cầu sau:

- ✓ Chức năng tra cứu số điện thoại (theo tên) và các thông tin cơ bản người dùng.
- ✓ Người sử dụng có thể thêm, sửa, xóa danh bạ điện thoại.
- ✓ Import dữ liệu từ file với định dạng cho trước.

## **2. Cơ sở lý thuyết:**

### **2.1. Cây đồ đen (thư viện JRB-libfdr):**

#### **2.1.1. Định nghĩa:**

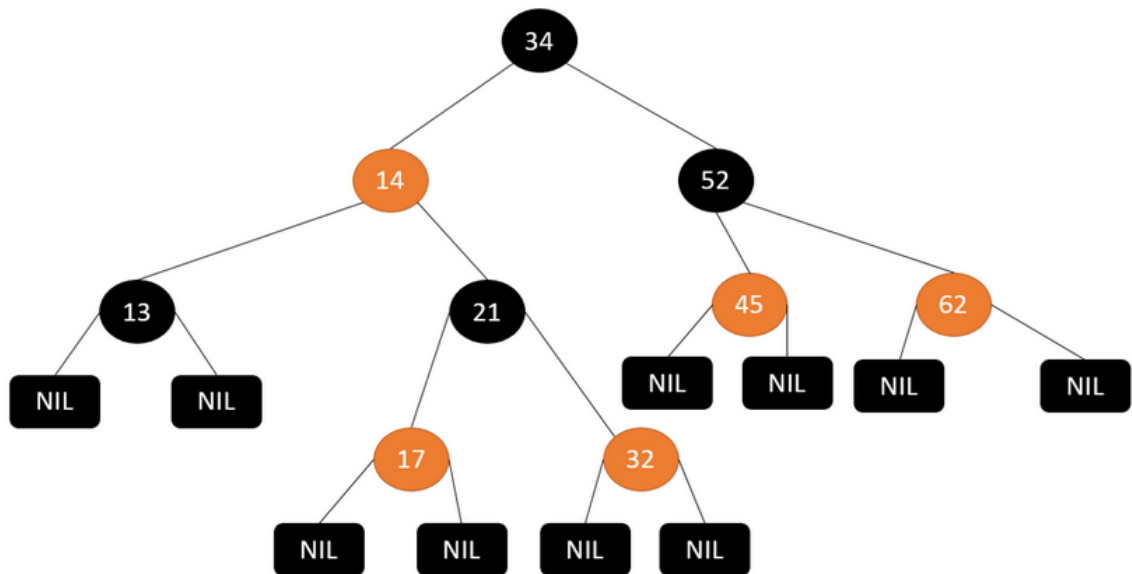
**Cây đồ đen** là một dạng cây tìm kiếm nhị phân tự cân bằng. Cấu trúc ban đầu của nó được đưa ra vào năm 1972 bởi Rudolf Bayer. Nó là cấu trúc phức tạp nhưng cho kết quả tốt về thời gian trong trường hợp xấu nhất.

#### **2.1.2. Tính chất:**

Mỗi nút của cây đồ-đen có thuộc tính “màu” nhận một trong hai giá trị “đỏ” hoặc “đen”, trong đó:

1. Một nút hoặc là đỏ hoặc là đen.
2. Gốc là đen.
3. Tất cả các lá (NULL) là đen.

4. Cả hai con của mọi nút đỏ là đen. (và suy ra mọi nút đỏ có nút cha là đen)
5. Tất cả các đường đi từ một nút bất kỳ tới các lá có số nút đen bằng nhau.



Ví dụ về cây đỏ đen

### 2.1.3. Thời gian thực hiện:

Các phép toán trên cây đỏ đen như tìm kiếm (search), chèn (insert), và xóa (delete) có độ phức tạp là  $O(\log n)$ , trong đó  $n$  là số các phần tử của cây.

### 2.1.4: Thư viện sử dụng :

libfdr Library:

[http://web.eecs.utk.edu/~huangj/cs360/360/notes/Libfdr/?fbclid=IwAR0lc\\_jQoRDaoSh8nJxoQkqfGF3v-S4pq0DPb5QuUJzj\\_ijkODEF5lFamMQ](http://web.eecs.utk.edu/~huangj/cs360/360/notes/Libfdr/?fbclid=IwAR0lc_jQoRDaoSh8nJxoQkqfGF3v-S4pq0DPb5QuUJzj_ijkODEF5lFamMQ)

Tác giả: Jim Plank

### Đánh giá:

Theo đánh giá của nhóm sau khi cài đặt chương trình danh bạ bằng thư viện này thì đây là một thư viện tốt:

- ✓ Cấu trúc thư viện rõ ràng

- ✓ Chương trình chạy nhanh
- ✓ Dễ cài đặt

**Cách sử dụng library (API Reference Manual):**

- Include:

*#include "libfdr/jrb.h"*

- Tạo một JRB\_Tree mới

*make\_jrb();*

- Giải phóng JRB-Tree

*jrb\_free\_tree(JRB root);*

- Tìm kiếm node:

*jrb\_find\_str(JRB n, char \*key);*

- Duyệt toàn bộ cây:

*jrb\_traverse(ptr, lst);*

- Thêm node vào B-Tree:

*jrb\_insert\_str(JRB tree, char \*key, Jval val)*

- Xóa node trong B-Treeb:

*jrb\_delete\_node(JRB node);*

## **2.2. GTK:**

Ở chương trình danh bạ này, chúng em sử dụng GTK+ viết bằng ngôn ngữ C kết hợp với Glade để tạo giao diện.

### **2.2.1. Tổng quát:**

**GTK+** là bộ công cụ rất hiệu quả để tạo một giao diện đẹp, có tính tương thích cao và dễ sử dụng cho chương trình, ứng dụng. **GTK+** được viết bằng ngôn ngữ C, nhưng có thể liên kết với rất nhiều các ngôn ngữ lập trình khác như ngôn ngữ C++, Python, C# v.v...

Mô tả ngắn gọn thì **GTK+** là một thư viện, nó tối giản việc tạo giao diện người dùng bằng đồ họa bằng cách cung cấp sẵn một tập các thành phần (component) được gọi là widgets. Cách tiếp cận này cho phép việc phát triển ứng dụng rất dễ dàng và nhanh chóng.

### **2.2.2. Cài Đặt GTK+ trong Ubuntu**

- Cài đặt online ta dùng lệnh:

```
sudo apt-get install libgtk3.0-dev
```

- Cài đặt libgtk2.0-dev ,libglib2.0-dev....,những thư viện cần thiết lập cho lập trình **GTK+**

```
sudo apt-get install pkg-config
```

- pkg-config được dùng để biên dịch các **GTK+** procedures và tự động xác định vị trí của các thư viện và head file

```
sudo apt-get install devhelp
```

- Cài đặt tài liệu hướng dẫn **GTK+** devhelp (rất hữu ích)

```
sudo apt-get install libglib2.0-doc libgtk2.0-doc
```

- Cài API reference manuals và các tài liệu trợ giúp cho gtk/glib

```
sudo apt-get install glade libglade2-dev
```

```
sudo apt-get install glade
```

### 2.3. Glade:

**Glade** là công cụ thiết kế giao diện người dùng cho **GTK+** và Gnome.

**Glade** hỗ trợ cho nhiều ngôn ngữ lập trình khác nhau như C, C++, Java, Perl, Python, C#, Pike, Ruby, Haskell, Objective Caml and Scheme... một cách dễ dàng.

**Glade** được sử dụng để thiết kế nguyên mẫu cho **GTK+** và Gnome một cách nhanh chóng. Nó cung cấp một khung chính và có thể thêm bớt hoặc sửa chữa các widget. Việc làm này rất đơn giản nhưng rất mạnh có thể giúp những người mới lập trình có thể thiết kế được giao diện người dùng trong thời gian ngắn.

Giao diện người dùng được **Glade** thiết kế dưới dạng XML. Bằng cách sử dụng thư viện Glibglade, người sử dụng có thể bỏ qua nhiều câu lệnh mà vẫn duy trì được giao diện chương trình. Glibglade thực sự hữu ích vì XML file được nạp vào theo chế độ chạy thực nên cho phép chỉnh sửa giao diện mà không phải dịch lại.

### 2.4. Thư viện Cgen:

**Cgen** là thư viện được xây dựng và phát triển bởi thầy Nguyễn Bá Ngọc, để triển khai khái quát các cấu trúc dữ liệu cơ bản như vector, stack, queue, ...

Làm sao để sử dụng thư viện trong ứng dụng?

Vì thư viện được thiết kế chỉ bao gồm các tệp tiêu đề sao cho chèn vào để sử dụng luôn. Trong ứng dụng có sử dụng vector và khai báo:

```
#include "cgen/svector.h"
```

➤ Tạo vector

```
vector_t gtv_create();
```

➤ Gán giá trị theo chỉ số của vector

```
gtv_set_value_with_index(vector_t *v, gtype value, long idx);
```

- Đặt lại size của vector

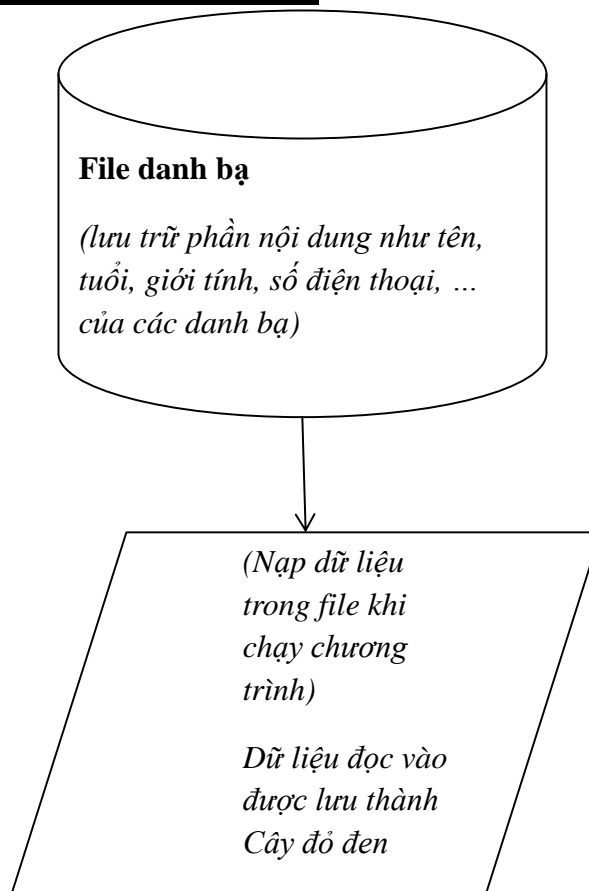
*gtv\_setsize();*

## **3. Thiết kế phần mềm:**

### **3.1. Chức năng:**

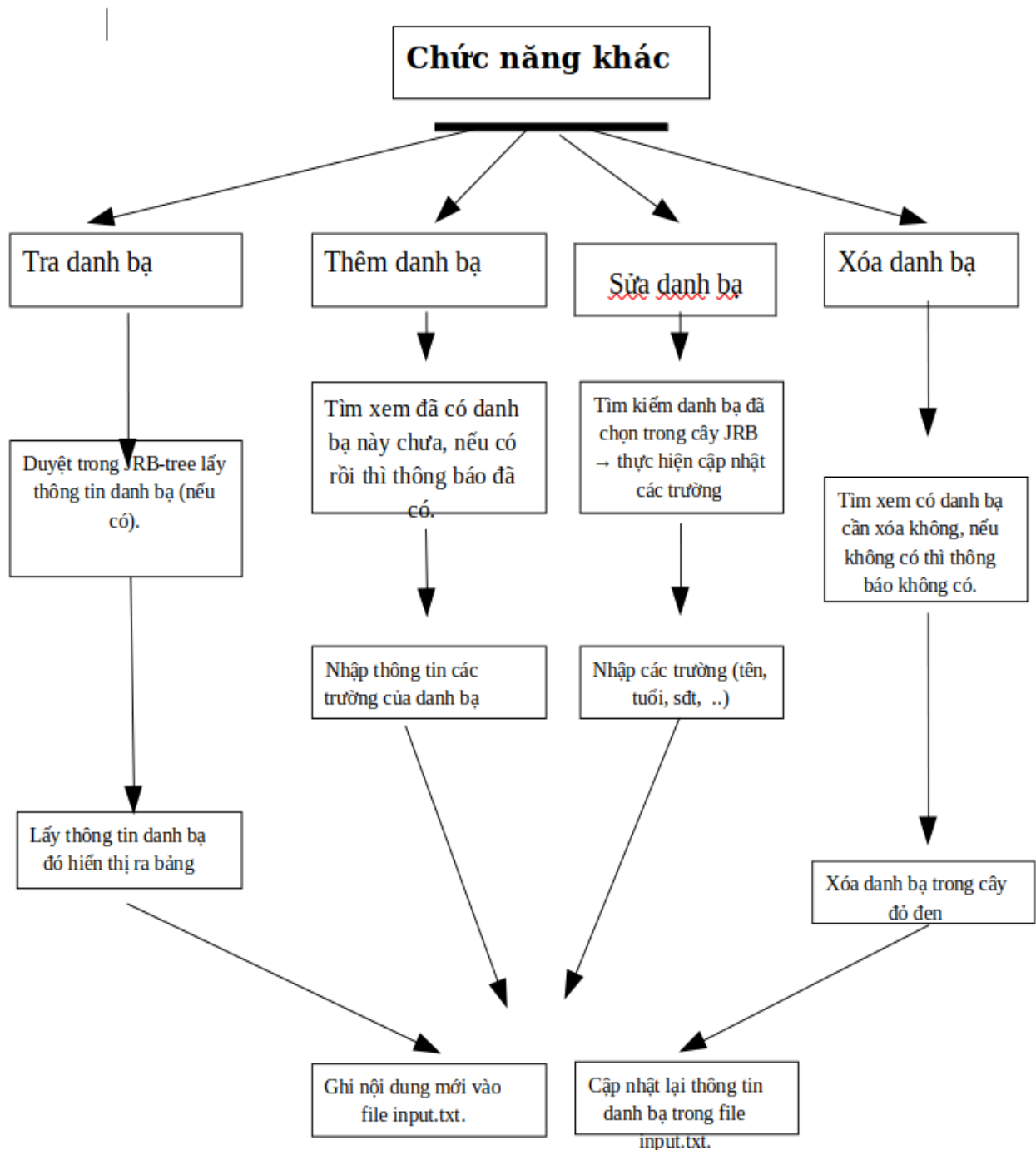
- ✓ Tra danh bạ (theo tên)
- ✓ Đọc dữ liệu danh bạ từ CSDL (lưu dưới dạng file .txt)
- ✓ Sửa thông tin danh bạ
- ✓ Xóa danh bạ
- ✓ Thêm danh bạ mới
- ✓ Hiển thị danh bạ

### **3.2. Tổ chức bộ nhớ lưu trữ:**





### 3.3. Quy trình thực hiện:



### 3.4. Thiết kế giao diện và vận hành:

#### *Thiết kế giao diện:*

- Sử dụng nền Ubuntu.
- Sử dụng GTK+ và Glade.



Giao diện chính



Chức năng sửa danh bạ

c\_project

新しい連絡先情報  
New Contact Information



連絡先 (Name): 名前

電話番号 (Telephone): 10桁にする必要があります

性 (Sex): man/womanでなければなり...

年齢 (Age): 何歳


メール (Mail): メールアドレス

アバター (Avatar file) 画像ファイルのパス

Choose Avatar Add Cancel

c\_project

連絡先情報を編集する  
Edit contact information



連絡先 (Name): Lê Văn Duẩn

電話番号 (Telephone): 1234567890

性 (Sex): man

年齢 (Age): 12

メール (Mail): duan.lv@gmail.com

アバター (Avatar file) /home/duan0308/C\_project/im.

Choose Avatar Apply Cancel

Chức năng thêm danh bạ

### ***Vận Hành:***

- Chạy trên nền Ubuntu 14.04
- Cmake version 2.6, ninja
- Phiên bản GTK+ 3.0
- Glade 3.22.2

## **4. Phân công và kế hoạch công việc:**

STT	Công Việc	Người Làm
1	Tìm hiểu, phân tích yêu cầu bài toán	Cả nhóm

2	Xây dựng giao diện	Cả nhóm
3	Dữ liệu đầu vào, code giao diện	Đinh Việt Long Nguyễn Hải Dương
4	Code các hàm nhận tín hiệu, các hàm xử lý dữ liệu bên trong chương trình	Lê Văn Duẩn
5	Test các testcase	Đinh Việt Long Nguyễn Hải Dương
6	Debug	Lê Văn Duẩn
7	Mô tả app và các chức năng của app	Đinh Việt Long
8	Báo cáo bài tập lớn	Nguyễn Hải Dương

## **5. Tài liệu tham khảo:**

- ❖ Bài giảng trên lớp
- ❖ Giáo Trình: Introduction to Algorithms (Third Edition) - Thomas H.Cormen (Giáo trình thầy Nguyễn Bá Ngọc gửi)
- ❖ libfdr: <https://github.com/bangoc/libfdr.git>
- ❖ Cgraph: <https://github.com/bangoc/cgraph.git>
- ❖ Cgen: <https://github.com/bangoc/cgen.git>
- ❖ GTK: <http://www.gtk.org/>