**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

A picture containing text, sign

Description automatically generated

BÁO CÁO PROJECT I

**CÁC THUẬT TOÁN NÉN NGUỒN RỜI RẠC**

**Chủ đề 1: Thuật toán mã hóa của Huffman**

**GVHD: PGS.TS Đặng Văn Chuyết**

|  |  |  |
| --- | --- | --- |
| **Thành viên nhóm** | **MSSV** | **Lớp** |
| Lê Minh Việt Anh | 20210037 | IT2-05-K66 |
| Nguyễn Mạnh Chiến | 20210118 | IT2-03-K66 |
| Nguyễn Trọng Nhật | 20215625 | IT2-05-K66 |
| Lê Việt Quang | 20215630 | IT2-05-K66 |

**Hà Nội, tháng 01 năm 2024**

**MỤC LỤC**

[1. Giới thiệu đề tài 3](#_Toc156025322)

[2. Phân tích đề tài 4](#_Toc156025323)

[2.1. Thuật toán mã hóa của Huffman 4](#_Toc156025324)

[2.2. Phân tích yêu cầu đề tài 4](#_Toc156025325)

[2.3. Phân công nhiệm vụ 4](#_Toc156025326)

[3. Kết quả 5](#_Toc156025327)

[3.1. Xây dựng chương trình 5](#_Toc156025328)

[3.1.1. Xử lý đầu vào, tạo mã Huffman từ nguồn tin (Việt Anh) 5](#_Toc156025329)

[3.1.2. Mã hóa bản tin (Nhật) 9](#_Toc156025330)

[3.1.3. Giải bản mã (Quang) 10](#_Toc156025331)

[3.1.4. Xử lý đầu vào, ra và giao diện (Chiến) 12](#_Toc156025332)

[3.2. Xây dựng trường hợp kiểm thử 17](#_Toc156025333)

[3.3. Kết quả của chương trình qua trường hợp kiểm thử 17](#_Toc156025334)

# 1. Giới thiệu đề tài

Mã hóa Huffman là một thuật toán mã hóa mạnh mẽ và phổ biến trong lĩnh vực khoa học máy tính và truyền thông. Được phát triển bởi David A. Huffman vào năm 1952, thuật toán này được sử dụng để nén dữ liệu với mục tiêu giảm dung lượng cần thiết để lưu trữ hoặc truyền tải thông tin. Phương pháp mã hóa Huffman dựa trên nguyên tắc gán mã ngắn hóa cho các ký tự xuất hiện thường xuyên hơn, giảm độ dài của mã binh thường so với mã cố định.

Thuật toán Huffman là một ví dụ điển hình của mã hóa không mất thông tin, nơi chúng ta chủ động giảm kích thước của dữ liệu mà không làm thay đổi nội dung thông tin. Bằng cách sử dụng cây Huffman, mỗi ký tự trong tập hợp dữ liệu được biểu diễn bằng một mã nhị phân duy nhất, với các mã ngắn hóa cho các ký tự xuất hiện phổ biến nhất và các mã dài hơn cho các ký tự xuất hiện ít.

Ứng dụng của mã hóa Huffman không chỉ giới hạn trong lĩnh vực lưu trữ dữ liệu, mà còn mở rộng ra nhiều lĩnh vực khác như truyền thông, nén hình ảnh, âm thanh và video. Đối với những ứng dụng yêu cầu hiệu quả cao và tiết kiệm băng thông, mã hóa Huffman đóng vai trò quan trọng trong việc tối ưu hóa kích thước dữ liệu và tăng tốc quá trình truyền tải. Đồng thời, việc nắm vững thuật toán này giúp lập trình viên hiểu rõ về cơ sở lý thuyết của mã hóa và giải mã, đồng thời có thể ứng dụng nó vào các ứng dụng thực tế.

# 2. Phân tích đề tài

## 2.1. Thuật toán mã hóa của Huffman

Cho nguồn tin X = {x1,x2,..,xq} có phân bố xác suất P(X) = (p1,p2,..,pq} được nhập từ bàn phím. Thuật toán tim bộ mã Hufman với cơ số mã r có các ký hiệu mã có giá trị từ 0 đến r-1 gồm các bước (tìm các từ mã ứng với mỗi tin của nguồn):

Bước 1: Tính số nguyên a = (q-r)/(r-1). Nếu a không nguyên thì tính giá trị mới của a’ = Int(a) +1. Tiếp theo tính số tin mới của nguồn q’ = r+ a’(r-1). Số tin ảo có xác suất bằng 0 là q’-q. Nếu a nguyên thì q’ = q.

Bước 2: Xếp các tin của nguồn q’ tin theo thứ tự xác suất giảm dân

Bước 3: Nhóm r tin cuối của nguồn thành 1 tin phụ có xác suất bằng tổng xác suất của các tin được nhóm vào. Đánh dấu mỗi tin được nhóm vào bằng một ký hiệu mã khác nhau.

Bước 4: Lặp lại các bước b và c cho đến khi nguồn còn đúng r tin. Đánh dấu mỗi tin của nguôn này bằng 1 ký hiệu mã khác nhau.

Bước 5: Từ mã ứng với mỗi tin là chuỗi các ký hiệu mã dùng đánh dấu chính tin này và các tin phụ chứa nó (nó được nhóm vào tin phụ này) theo thứ tự ngược với thứ tự nhóm

## 2.2. Phân tích yêu cầu đề tài

Đề tài bao gồm ba yêu cầu:

1. Xây dựng thuật toán mã hóa tin từ nguồn tin
2. Sử dụng bộ mã Huffman để mã hóa một bản tin
3. Sử dụng bộ mã Huffman để giải mã một bản mã

## 2.3. Phân công nhiệm vụ

|  |  |
| --- | --- |
| **Tên thành viên** | **Nhiệm vụ** |
| Lê Minh Việt Anh | Tạo mã Huffman từ nguồn tin, làm báo cáo |
| Nguyễn Mạnh Chiến | Thiết kế giao diện, làm báo cáo |
| Nguyễn Trọng Nhật | Giải mã bản mã, làm báo cáo |
| Lê Việt Quang | Mã hóa bản tin, làm báo cáo |

# 3. Kết quả

Chương trình sử dụng các biến tổng thể:

|  |  |  |
| --- | --- | --- |
| **Tên biến** | **Kiểu** | **Ý nghĩa** |
| base | int | Cơ số của mã Huffman |
| Code | unordered\_map<string, string> | Chứa nguồn tin và mã Huffman tương ứng (để mã hóa) |
| Source | nordered\_map<string, double> | Chứa nguồn tin, gồm ký hiệu và xác suất xuất hiện của chúng |
| reverseCode | unordered\_map<string, string> | Chứa mã Huffman và nguồn tin tương ứng (để giải mã) |
| exist | unordered\_map<string, bool> | Ghi nhận sự tồn tại của các tin của nguồn |

## 3.1. Xây dựng chương trình

### 3.1.1. Xử lý đầu vào, tạo mã Huffman từ nguồn tin (Việt Anh)

Input:

* Các cặp giá trị (s,p) trong đó s là giá trị của tin, p là xác suất xuất hiện tin đó
* Cơ số base của mã Huffman

Output:

* Các cặp giá trị (s, c) trong đó s là giá trị của tin, c là mã Huffman tương ứng

#### 3.1.1.1. Xử lý đầu vào

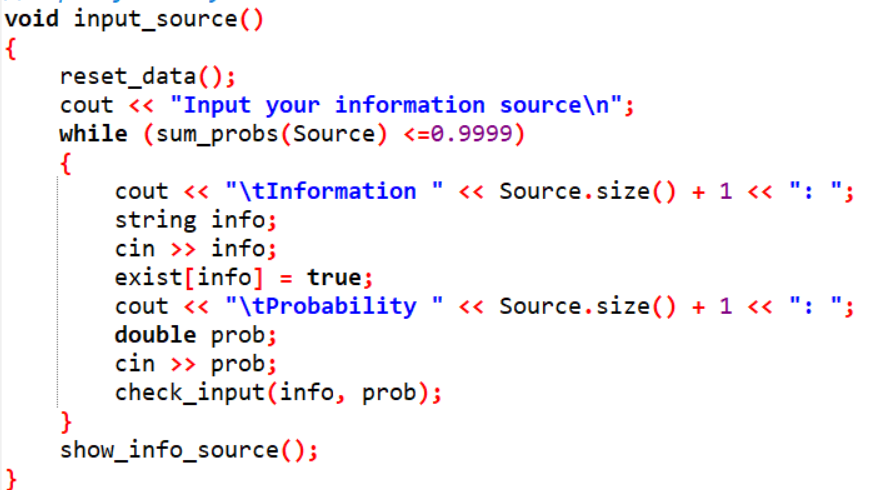
Trước khi nhập dữ liệu, ta sẽ xóa dữ liệu đã có trước đó (nếu tồn tại):

A black text on a white background

Description automatically generated

Ta kiểm soát phần đầu vào đảm bảo các điều sau:

* Kết thúc nhập khi tổng xác suất bằng 1
* Yêu cầu nhập lại một tin của nguồn khi tổng xác suất lớn hơn 1



Sau khi nhập, ta xem xét việc sinh mã Huffman từ nguồn hợp lệ.

#### 3.1.1.2. Xây dựng mã Huffman từ nguồn tin

Để thuận tiện, ta chuyển sang ngôn ngữ đồ thị dạng cây. Trong bài toán này là cây **base**-phân đầy đủ (là cây mà mỗi nút hoặc không có nút con nào, hoặc có đủ **base** nút con).Mỗi nút chứa cặp giá trị là tin (kiểu **string**) và xác suất xuất hiện của chúng. Ta xây dựng ngược cây base-phân đầy đủ như sau:

Các nút lá là cặp giá trị gồm tin của nguồn và xác suất xuất hiện của nó.

Từ **base** nút có xác suất nhỏ nhất ở độ cao i (i = 0, 1, 2, ...), tạo một nút mới là cha của các nút này, có giá trị là chuỗi ghép từ giá trị của các nút con, xác suất bằng tổng các xác suất của các nút con. Mỗi cạnh nối nút cha và nút con sẽ được gán một giá trị từ 0 đến **base-1** theo thứ tự xác định (từ nút con có xác suất nhỏ nhất đến nút con có xác suất nhỏ thứ **base**).

Đến cuối cùng ta được một nút duy nhất, là gốc của cây. Mã Huffman của mỗi tin (mỗi nút lá) được hình thành bởi một đường đi từ gốc đến nút lá chứa tin đó.

**Xây dựng chương trình từ thuật toán:**

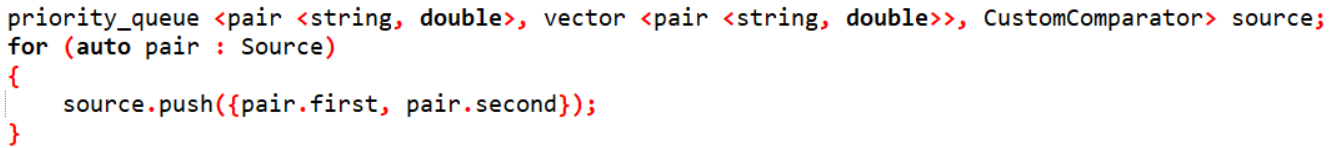
Đầu tiên, ta tạo thêm các tin ảo cho nguồn để có thể xây dựng được cây **base**-phân đầy đủ:

A white background with black text

Description automatically generated

Trong đó **base** là cơ số của mã Huffman, **num\_r\_info** là số tin thật, **num\_i\_info** là số tin ảo, **num\_newsource** là số tin của nguồn sau khi đã bổ sung các tin ảo.

Ta sử dụng hàng đợi ưu tiên giá trị nhỏ hơn, để thuận tiện cho việc tìm **base** giá trị nhỏ nhất, và **unordered\_map adj** để lưu trữ các nút kề của một nút trong đồ thị. Danh sách kề của một nút sử dụng kiểu dữ liệu **vector**, số phần tử của vector là 1 hoặc 0. Mối quan hệ kề của hai nút được định nghĩa: ***Nếu nút B là cha của nút A, ta thêm nút B vào danh sách kề của A.***



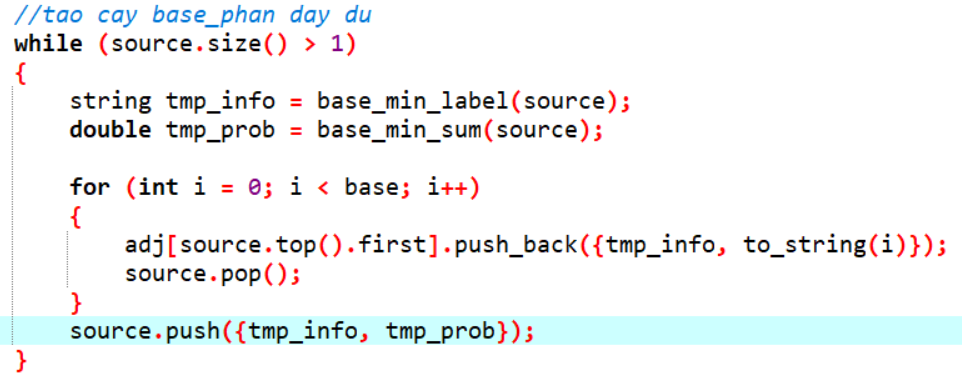
(Bổ sung các tin thật vào hàng đợi ưu tiên)

A group of black text

Description automatically generated with medium confidence

(Bổ sung các tin ảo, là các cặp giá trị (“i0”, 0), ..., (“ia”,0) vào hàng đợi ưu tiên)

Tiếp theo, ta xây dựng cây **base**-phân đầy đủ bằng cách thêm các nút mới vào danh sách kề của các nút. Đồng thời, các nút mới này được tạo được bổ sung vào hàng đợi ưu tiên, các nút con của nó sẽ bị xóa khỏi hàng đợi.



A screenshot of a computer code

Description automatically generated

(Các hàm **base\_min\_label**, **base\_min\_sum** dùng để tạo nhãn và xác suất cho các nút mới)

Cuối cùng, ta xây dựng mã Huffman bằng việc đi từ nút lá về gốc, thông qua danh sách kề:

A screenshot of a computer code

Description automatically generated

Tạo bảng **reverseCode** để sử dụng cho việc giải bản mã:

A black and red text

Description automatically generated

### 3.1.2. Mã hóa bản tin (Nhật)

Input:

* Bản tin sequence chứa các kí tự nằm trong bộ mã huffman đã được mã hóa

Output:

* Bản mã huffman\_sequence

Sử dụng biến string sequence – để chứa bản tin cần mã hóa; và biến string huffman\_sequence để chứa bản tin sau khi mã hóa.

Tạo thủ tục input\_sequence() nhập bản tin đầu vào từ bàn phím.

A white background with black text

Description automatically generated

Tuy nhiên, khi nhập bản tin đầu vào thì có thể nhập sai (trong bản tin có chứa kí tự không xuất hiện trong nguồn tin đã cho, nên phải tạo hàm để kiểm tra điều này). Trong quá trình nhập nguồn tin thì ghi nhận sự tồn tại của các kí tự bằng unordered\_map <string,bool> exist với trường [key] là kí tự đó và trường [value] là xác nhận sự tồn tại của nó ( true: đã tồn tại, flase: chưa tồn tại)

A computer screen shot of a code

Description automatically generated

Hàm check sẽ trả về true nếu trong bản tin nhập vào không chứa kí tự chưa được khai báo trong nguồn tin.

Thủ tục xử lý và tạo mã Huffman:

A computer screen shot of a code

Description automatically generated

Người dùng sẽ phải nhập đúng bản tin đầu vào, nếu không phải nhập lại. Sau khi nhập được bản tin đầu vào, chương trình sẽ lấy ra từng kí tự trong bản tin đó, ánh xạ sang từ mã, rồi nối vào chuỗi huffman\_sequence chứa các từ mã tạo thành bản mã

### 3.1.3. Giải bản mã (Quang)

A screenshot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

- Hàm giải mã với tham số đầu vào là mã huffman

- Hàm khởi tạo 1 chuỗi trống “str” sau đó duyệt qua từng ký tự trong chuỗi mã hufman và thêm ký tự đó vào chuỗi trống ấy

- Duyệt đến khi nào tìm thấy mã huffman tương ứng với mã “str” thì ta lưu ký tự tương ứng với mã huffman ấy vào 1 chuỗi “ string decodedMessage”, rồi trả chuỗi “str” về chuỗi trống.

- Sau đó lại tiếp tục duyệt tiếp đến khi kết thúc rồi in ra chuỗi sau khi giải mã thành công.

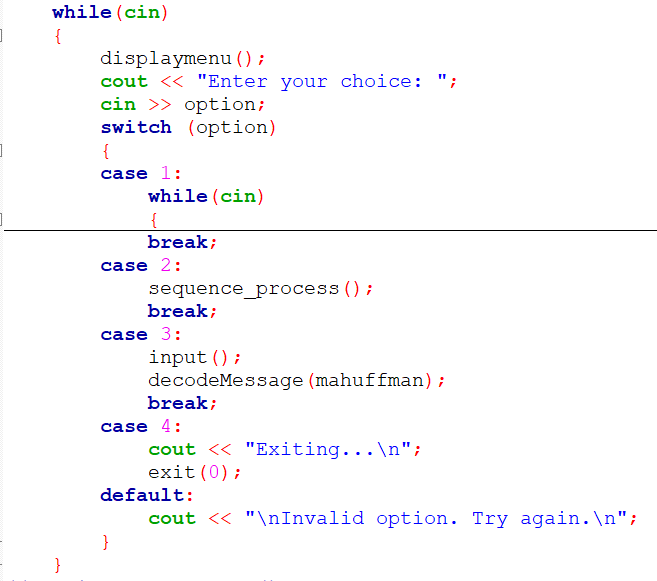
- Nếu duyệt đến cuối mà str vẫn còn giá trị trong đó thì yêu cầu “Nhap lai ma huffman” sẽ được hiển thị.

### 3.1.4. Xử lý đầu vào, ra và giao diện (Chiến)

A screenshot of a computer program

Description automatically generated

* Menu:
  + 1. Tạo bảng mã Huffman
  + 2. Mã hoá
  + 3. Giải mã
  + 4. Thoát chương trình
* Menu tạo bảng mã có 2 kiểu nhập vào dữ liệu
  + 1. Dữ liệu nhập vào từ bàn phím
  + 2. Dữ liệu nhập vào từ 1 file
  + 3. Trở về Menu ban đầu



A black screen with white text

Description automatically generated

- Hàm sẽ lặp đi lặp lại cho đến khi người dùng lựa chọn 4 thoát chương trình

#### 3.1.4.1. Tạo mã Huffman

- Đối với lựa chọn đầu tiên, người dùng sẽ được đưa tới menu lựa chọn phương thức nhập

A screen shot of a computer code

Description automatically generatedA black background with white text

Description automatically generated

- Tương tự như trên, menu này sẽ lặp lại cho đến khi người dùng chọn lựa chọn 3 là trở về menu ban đầu.

- Nếu người dùng lựa chọn phương thức nhập từ bàn phím, thì sẽ nhập vào information sau đó đến probability cho đến khi tổng probability bằng 1, sau đó sẽ nhập vào base. Khi thực hiện xong chương trình sẽ bắt đầu tạo mã Huffman từ dữ liệu nhập vào.

- Ngoài ra chương trình còn tính toán thời gian thực hiện và in ra cùng với bảng mã.

A screenshot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

- Người dùng lựa chọn phương thức nhập vào từ file thì sẽ phải nhập vào tên của file và trong file sẽ có định dạng như information 1 probability 1 information 2 probability 2 ...:

A black background with white text

Description automatically generated

- Khi đó chương trình sẽ tiến hành kiểm tra và thực hiện như phương thức thứ nhất

- Trong trường hợp file không tồn tại thì sẽ in ra lỗi và yêu cầu nhập lại

A black background with white text

Description automatically generated

#### 3.1.4.2. Mã hoá



- Người dùng sẽ được yêu cầu nhập vào xâu kí tự cần mã hoá

- Nếu trong đó chứa kí tự chưa được mã hoá thì sẽ thông báo lỗi



- Nếu không thì sẽ trả về mã hoá của dữ liệu nhập vào

#### 3.1.4.3. Giải mã

A black background with white text

Description automatically generated

- Người dùng sẽ được yêu cầu nhập vào một đoạn mã hoá

- Chương trình sẽ thực hiện giải mã nó và trả về xâu kí tự đã được giải mã.

- Nếu mã nhận vào không hợp lệ thì chương trình sẽ yêu cầu nhập lại.

## 3.2. Xây dựng trường hợp kiểm thử

|  |  |  |
| --- | --- | --- |
| Input | Expected Output | Result |
| 1 | Hiển thị menu nhập nguồn tin | OK |
| 1 a 0.5 b 0.3 c 0.15 d 0.05 | Hiển thị bảng cho nguồn tin và xác suất | OK |
| 2 | Hiển thị bảng mã Huffman | OK |
| 3 | Quay lại menu chính | OK |
| 2 abcdcba | 011101100101110 | OK |
| 3 011101100101110 | abcdcba | OK |
| 4 | Kết thúc chương trình | OK |

## 3.3. Kết quả của chương trình qua trường hợp kiểm thử

Qua một số trường hợp kiểm thử, chương trình cho ra kết quả đúng với mong đợi. Thuật toán được cài đặt thành công. Dưới đây là một số thống kê:

* Số dòng lệnh: 388
* Số hàm được dùng: 19
* Kích thước chương trình: 11 KB

--------------HẾT--------------