**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**
**TRƯỜNG ĐIỆN – ĐIỆN TỬ**
---o0o---


# BÁO CÁO MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT (ET2100)

## ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH NÉN VÀ GIẢI NÉN FILE BẰNG THUẬT TOÁN HUFFMAN

Giảng viên hướng dẫn: Thầy Đào Trung Kiên

Mã lớp: 133322

Nhóm sinh viên thực hiện: Nhóm 5

| Họ tên | MSSV |
|---|---|
| Nguyễn Đình Phúc | 20203530 |
| Lê Trung Kiên | 20203474 |
| Trần Sỹ Hoàng | 20203435 |
| Phạm Mạnh Dũng | 20203383 |

***Hà Nội, tháng 7 năm 2022***

## 1.Giới thiệu về thuật toán Huffman

### Tác giả

Thuật toán được đề xuất bởi David A.Huffman khi ông còn là sinh viên Ph.D.tại MIT, và công bố năm 1952 trong bài báo "A Method for the Construction of Minium-Redundancy Codes".Sau này Huffman đã trở thành một giảng viên ở MIT và sau đó ở khoa Khoa học máy tính của Đại học California, SantaCruz, Trường Kỹ nghệ Baskin( Baskin School of Engineering).

### Thuật toán Huffman

Các tập tin của máy tính được lưu dưới dạng các kí tự có chiều dài không đổi là 8 bits. Trong nhiều tập tin, xác suất xuất hiện các kí tự này là nhiều hơn các kí tự khác, từ đó ta thấy ngay rằng nếu chỉ dùng một vài bit để biểu diễn cho các kí tự có xác suất xuất hiện lớn và dùng nhiều bit hơn để biểu diễn cho các kí tự có xác suất xuất hiện nhỏ thì có thể tiết kiệm được độ dài tập tin một cách đáng kể. Ví dụ, để mã hoá một chuỗi như sau:

"ABRACADABRA"

Nếu mã hoá chuỗi trên trong dạng mã nhị phân 5 bit ta sẽ có dãy bit sau :

00001000101001000001000110000100100000010001010010000001

Để giải mã thông điệp này, chỉ đơn giản là đọc ra 5 bits ở từng thời điểm và chuyển đổi nó tương ứng với việc mã hoá nhị phân đã được định nghĩa ở trên. Trong mã chuẩn này, chữ D xuất hiện chỉ một lần sẽ cần số lượng bit giống chữ A xuất hiện nhiều lần.

Ta có thể gán các chuỗi bit ngắn nhất cho các kí tự được dùng phổ biến nhất, giả sử ta gán: A là 0, B là 1, R là 01, C là 10 và D là 11 thì chuỗi trên được biểu diễn như sau: 0 1 01 0 10 0 11 0 1 01 0 Ví dụ này chỉ dùng 15 bits so với 55 bits như ở trên, nhưng nó không thực sự là một mã vì phải lệ thuộc vào khoảng trống để phân cách các kí tự. Nếu không có dấu phân cách thì ta không thể giải mã được thông điệp này. Ta cũng có thể chọn các từ mã sao cho thông điệp có thể được giải mã mà không cần dấu phân cách, ví dụ như: A là 11, B là 00, C là 010, D là 10 và R là 011, các từ mã này gọi là các từ mã có tính prefix (Không có từ mã nào là tiền tố của từ mã khác). Với các từ mã này ta có thể mã hoá thông điệp trên như sau:

11000111101011101100011111

Với chuỗi đã mã hoá này ta hoàn toàn có thể giải mã được mà không cần dấu phân cách. Nhưng bằng cách nào để tìm ra bảng mã một cách tốt nhất ?

Trong khoa học máy tính và lý thuyết thông tin, mã Huffman là một thuật toán mã hóa dùng để nén dữ liệu. Nó dựa trên bảng tần suất xuất hiện các kí tự cần mã hóa để xây dựng một bộ mã nhị phân cho các kí tự đó sao cho dung lượng (số bít) sau khi mã hóa là nhỏ nhất.

**2. Phân chia công việc**

| Công việc | Người thực hiện |
|---|---|
| Xây dựng thuật toán và viết code nén file | Nguyễn Đình Phúc, Lê Trung Kiên |
| Xây dựng thuật toán và viết code giải nén file | Trần Sỹ Hoàng, Phạm Mạnh Dũng |

**3. Giải thích các thuộc tính và phương thức của các class:**

**class Node:**

*thuộc tính:

```
int data;//ký tự theo mã ascii
int freq;//tần số của ký tự
Node* pLeft; //Nút trái
Node* pRight;//Nút phải
```

*phương thức:

```
static Node* createdNode(int _data, int _freq);//Tạo nút lá từ ký tự và tần số
static Node* createdRoof(int _freq, Node* right, Node* left);//Tạo nút từ tổng tần số và 2 nút trái-phải
```

**class Encoding:**

*thuộc tính:

```
string out_name;   //tên file được nén
string file_name;  //tên file cho vào nén
string row_of_bits; //biến đổi chuỗi cho vào thành mã nhị phân
ascii
int* array; //lưu các ký tự của file
int num;
Node* Tree;       //xử lý cây huffman.
map<int, string> codes;  //lưu lại đường đi của ký tự, sử dụng
kiểu 'int' tương ứng trong bảng mã ascii.
map<int, int> freqs;   //lưu lại tần số của mỗi ký tự
```

*phương thức:

```
Encoding();//Hàm khởi tạo không đối số
Node* get_Tree();//Lấy cây Huffman
void Input_filename();//Nhập tên file vào ra
void Set_filename(string, string);//Lấy tên file vào ra
void save_Lines();//Lưu từng ký tự của file vào mảng ký tự
void countFreq();//Hàm đếm tần suất của ký tự trng mã ascii, từ 0
đến 255
void flip_map();//Hỗ trợ xây dựng cây Huffman
void BuildArray_Huffman_node(vector<Node*>&);//Xây dựng cây
Huffman
void printCodes(Node*, string str = "");//In đường đi của mỗi ký
tự
void save_rowofbit();//Lưu mã bit của dữ liệu file
void output_file();//Xuất ra file kết quả nén
void out_file_txt(vector<Node*> tree);//Lưu vào file các thông số
phục vụ cho việc giải nén
```

**class Decoding:**

*thuộc tính:

```
string out_name;//tên file chứ kết quả giải nén
string file_name;//tên file cần giải nén
char* row_of_bits;//Lưu chuỗi bit
int* array;//lưu ký tự
int num;//số ký tự
Node* Tree;//Lưu cây Huffman
map<int, string> codes;//lưu lại đường đi của ký tự, int đại diện cho mã ascii
map<int, int> freqs;//lưu tần số của mỗi ký tự, int đại diện mã ascii cho kỹ tự
```

*phương thức:

```
Decoding();//Hàm tạo không đối số
Node* get_Tree();//Lấy cây Huffman
void Input_filename();//Nhập các tên file
void Set_filename(string, string);//Lưu tên file
void flip_map();//Hỗ trợ xây dựng cây
void BuildArray_Huffman_node(vector<Node*>&);//Xây cây Huffman
void printCodes(Node*, string str = "");//Lấy đường đi
void save_rowofbit();//Lưu chuỗi bit
void output_file();//xuất kết quả vào file
void countFreq_txt();//Lấy dữ liệu từ file để xây cây
```

**4. Thuật toán nén**

Dùng thuật toán nén Huffman tĩnh như sau :

**Bước 1:** Duyệt file, đọc từng byte lưu vào mảng F

Ví dụ chuỗi kỹ tự cần nén:

F = "ABABBCBBDEEEABABBAEEDDCCABBBCDEEDCBCCCCDBBBCAAA"
N=47

**Bước 2:** Đếm tần số của mỗi kí tự ( mã ASCII) xuất hiện trong mảng F -> ta có bảng tần số

Bảng tần suất xuất hiện

| Ký tự | Tần suất |
|---|---|
| A | 9 |
| B | 15 |
| C | 10 |
| D | 6 |
| E | 7 |

**Bước 3:** Xây dựng cây Huffman dựa vào bảng tần số

-B1: Sắp xếp bảng tần suất theo thứ tự tần suất giảm dần.

-B2: Sau đó tạo các node, mỗi node đều chứa dữ liệu một ký tự và tần suất của ký tự đó.

-B3: Lặp lại thao tác sau cho đến khi chỉ còn một node:

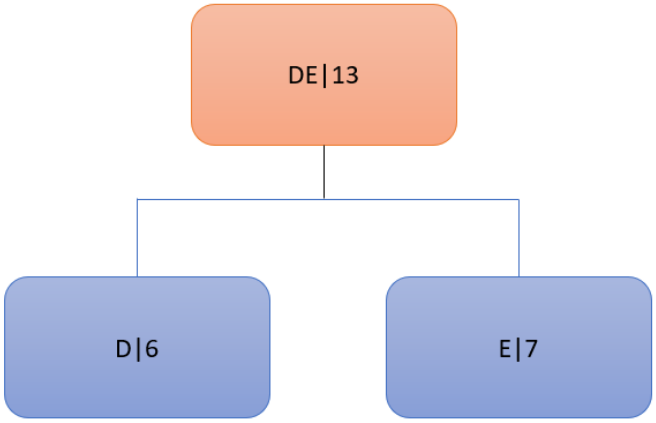
+ Tạo node cha trỏ đến 2 node có tần suất nhỏ nhất, có tần suất bằng tổng 2 node con.

+ Chèn node cha vừa tạo vào vị trí thích hợp theo thứ tự tần suất.

+ Xóa 2 node vừa bị trỏ đến.

Xây dựng cây Hufman

| Ký tự | Tần suất |
|---|---|
| A | 9 |
| B | 15 |
| C | 10 |
| D | 6 |
| E | 7 |

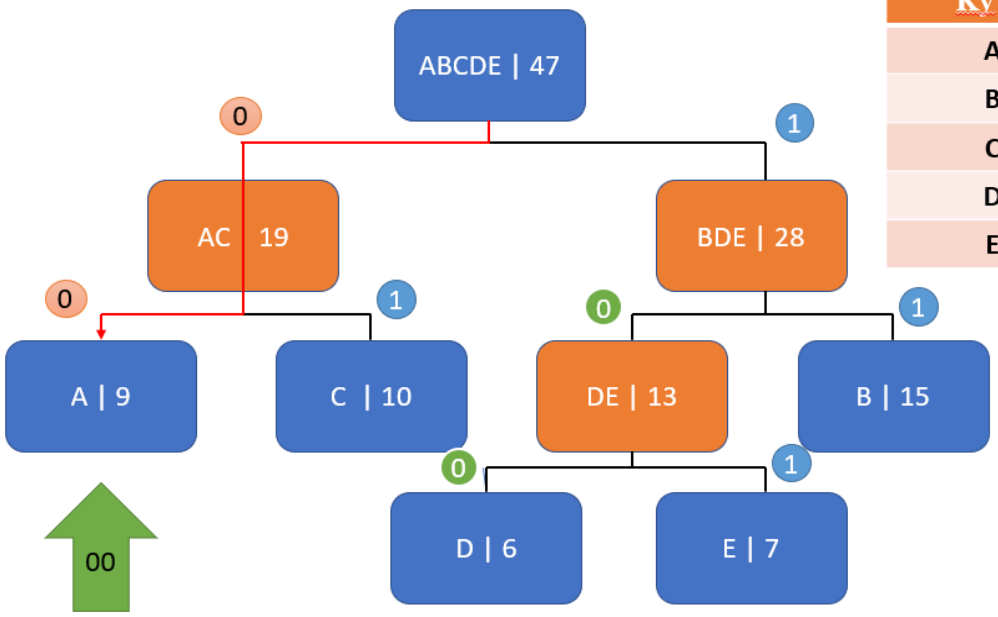



| Ký tự | Tần suất |
|---|---|
| B | 15 |
| DE | 10 |
| C | 10 |
| A | 9 |

| Ký tự | Tần suất |
|---|---|
| AC | 19 |
| B | 15 |
| DE | 13 |

| Ký tự | Tần suất |
|---|---|
| BDE | 28 |
| AC | 19 |

| Ký tự | Tần suất |
|---|---|
| ABCDE | 47 |




| Ký tự | Mã Huffman |
|---|---|
| A | 00 |
| B | 11 |
| C | 01 |
| D | 100 |
| E | 101 |

**Bước 4:** Chuyển các byte trong mảng F thành dãy 1 & 0 theo đường đi từ cây Huffman (qua phải 1 qua trái 0, sau đó lưu vào mảng Foutput - mảng chỉ chứa 0 & 1.

Chuỗi ký tự cần nén:

F = “ABABBCBBDEEEABABBAEEDDCCABBBCDEEDCBCCCCDBBBCAAA”

Sau khi nén, ta được chuỗi :

Foutput "00110011110111111001011011010011001111001011011001000101001111110110010110110001110101010110011111101000000"

**Bước 5:** Nếu số lượng bit ở Foutput không chia hết cho 8 thì mình thêm vào số kí tự 0 ít nhất để mảng Foutput chia hết cho 8, nếu chia hết thì giữ nguyên .

Chuỗi trên có độ dài là 107 nên cộng thêm vào chuỗi 5 kí tự ‘0’

**Bước 6:** Chuyển lần lượt 8 ký tự trong mảng Fout thành mã ASCII rồi xuất ra file, 8 ký tự tương đương 8bits-1byte.

Mã nhị phân: 00110011 chuyển thành mã thập phân là 51 - tương ứng với kí tự 3

Kết quả nén chuỗi trên:

3ß–Ó<¶E?e±Õgè

**5. Thuật toán giải nén**

Bước 1: Nhập tên file cần giải nén và tên file xuất ra

Bước 2: Duyệt file cần giải nén dưới dạng dãy nhị phân

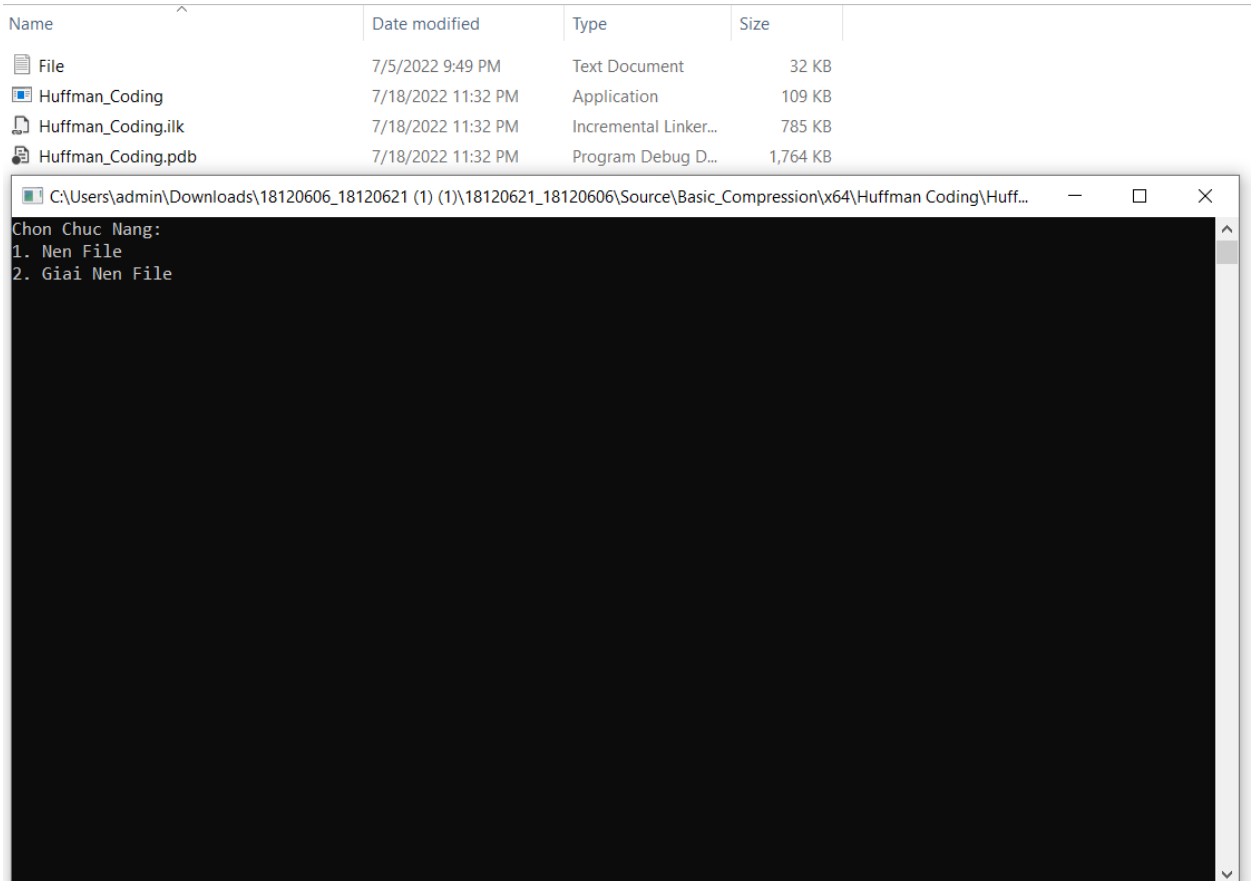
Bước 3: Đọc bảng tần suất và lưu từng byte đã mã hóa của file

Bước 4: Xây dựng cây Huffman từ bảng tần suất

Bước 5: Xuất file giải nén theo dãy bit đã mã hóa từ cây Huffman.

## 6. Hướng dẫn sử dụng

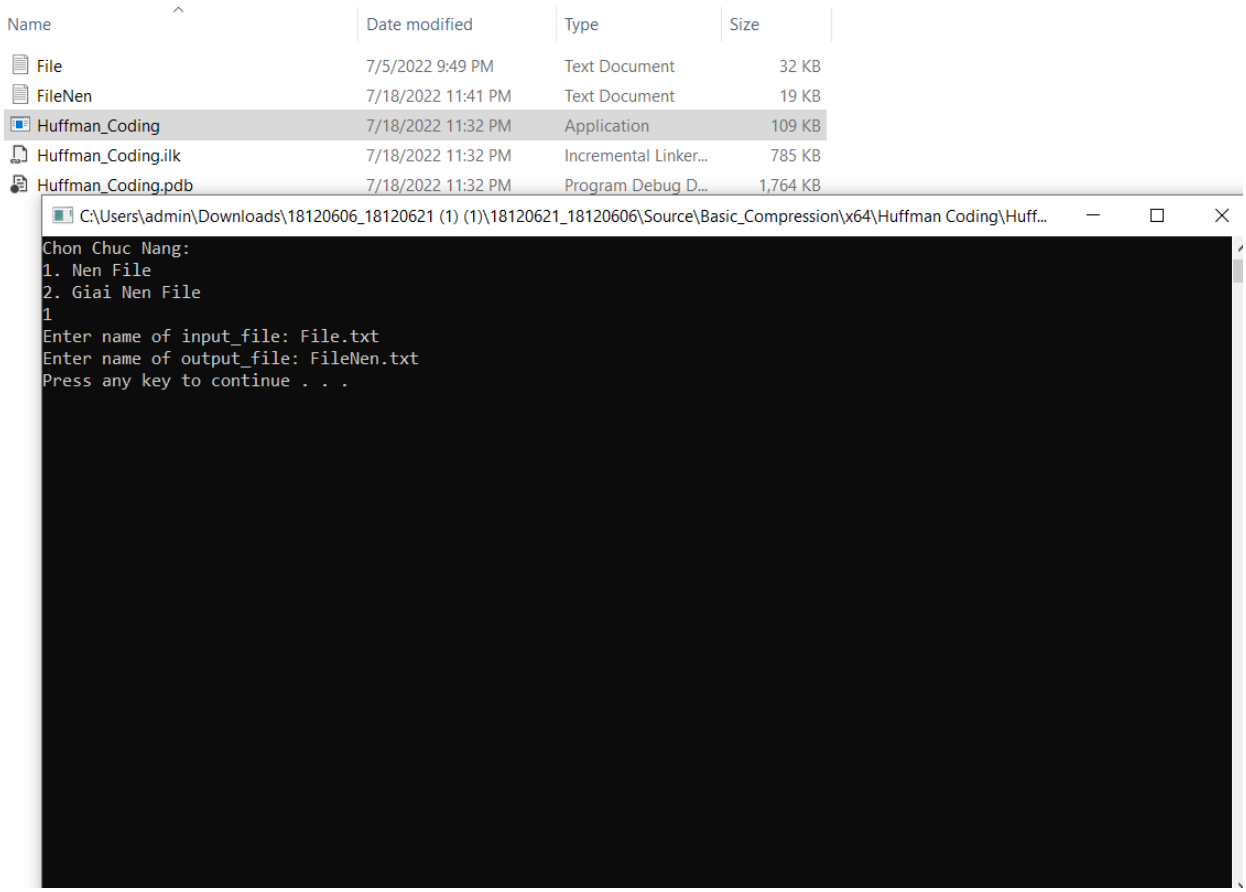
Sau khi khởi chạy chương trình, trên màn hình xuất hiện menu như hình 6.1



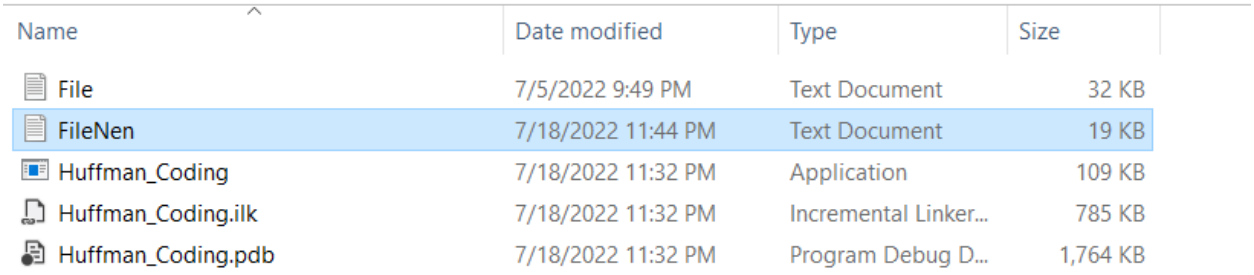

( **Hình 6.1**)

Người dùng lựa chọn thao tác muốn thực hiện trên menu:

1- Nhấn 1 để chọn nén file, 2 để giải nén file,
2- Ví dụ thao tác: Sau khi nhấn chọn 1, ta cần nhập vào tên file nguồn cần nén và tên file kết quả nén ( hình 6.2), ta nhận được kết quả là file sản phẩm trong thư mục gốc trước đó ( hình 6.3)




**Hình 6.2**




**Hình 6.3**

## 7. Video hướng dẫn

Link video: lv_0_20220719005556.mp4

## 8. Tài liệu tham khảo


1.https://www.programiz.com/dsa/huffman-coding

2.https://chidokun.github.io/2021/07/huffman-coding-p1/

3.https://web.stanford.edu/class/archive/cs/cs106x/cs106x.1192/resources/minibrowser2/huffman-encoding-supplement.pdf