

BÁO CÁO ĐỒ ÁN 2

Cấu trúc dữ liệu và giải thuật

1712078 - Ngô Phan Nhật Lâm

1712212 - Lý Thiên Ân



MỤC LỤC

Mức độ hoàn thành	3
Phân chia công việc	4
Mô tả các chức năng và cách thực hiện	5
Nén dữ liệu bằng thuật toán Huffman	5
Nén dữ liệu bằng thuật toán nâng cao	
Tham khảo & Video Demo	8

1

Mức độ hoàn thành

Nội dung	Thành phần	Mức độ hoàn thành
Nén tập tin bằng thuật toán Huffman	Nén tập tin chuỗi (.txt)	100%
	Nén tập tin chứa các kiểu dữ liệu còn lại (.cpp, .jpg,...)	100%
Nén tập tin bằng thuật toán nâng cao		

2 Phân chia công việc

MSSV	Họ Tên	Công việc thực hiện
1712078	Ngô Phan Nhật Lâm	<ul style="list-style-type: none">• Bài 1: Nén dữ liệu bằng thuật toán Huffman:<ul style="list-style-type: none">◦ Tập tin chuỗi (.txt)◦ Tập tin khác (.cpp, .jpg,...)
1712212	Lý Thiên Ân	<ul style="list-style-type: none">• Bài 2: Nén dữ liệu bằng thuật toán nâng cao:

3

Mô tả các chức năng và cách thực hiện

3.1 Nén dữ liệu bằng thuật toán Huffman:

- Khởi tạo cây nhị phân Huffman để chứa các ký tự (node) trong chuỗi.

```
struct NODE {
    unsigned char    c;        // ký tự
    int              freq;     // số lần xuất hiện
    bool             used;     // đánh dấu node đã xử lý chưa
    int              nLeft;    // chỉ số nút con nam bên trái
    int              nRight;   // chỉ số nút con nam bên phải
};

NODE    huffTree[MAX_NODE];
MABIT   bangMaBit[256];

void KhoiTao() {
    for (int i = 0; i < MAX_NODE; i++) {
        huffTree[i].c = i;
        huffTree[i].freq = 0;
        huffTree[i].used = false;
        huffTree[i].nLeft = -1;
        huffTree[i].nRight = -1;
    }
}
```

- Thống kê tần suất xuất hiện của các ký tự trong chuỗi. Tìm từng cặp phần tử xuất hiện ít nhất trong bảng thống kê. Xây dựng cây nhị phân dựa vào thuật toán Huffman. Lặp lại quá trình này cho đến khi còn 1 node chưa được xét thì dừng.

```
+void ThốngKêTầnSốXuấtHiện(char* tenFile) { ... }

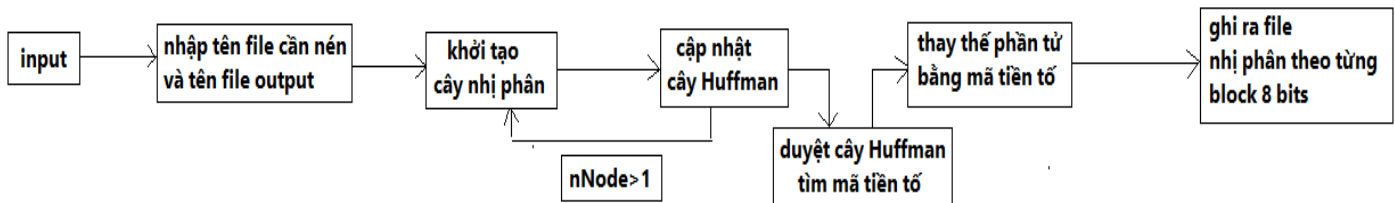
+void XuấtBảngThốngKê() { ... }

+bool Tìm2PhầnTửMin(int &i, int &j, int nNode) { ... }

+int TaoCâyHuffman() { ... }
```

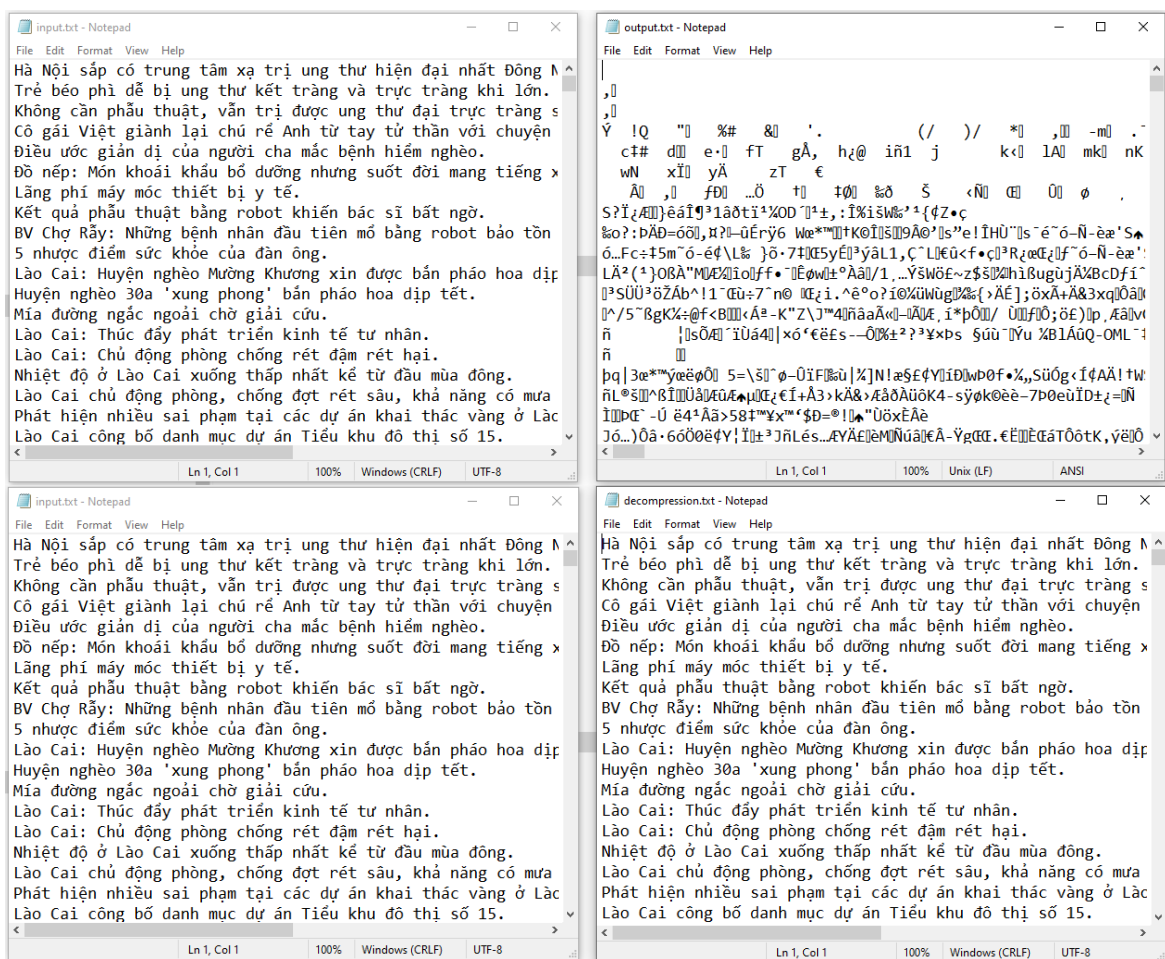
- Duyệt cây Huffman từ nốt gốc để tìm mã tiền tố cho mỗi ký tự theo quy ước rẽ trái gán bit 0, rẽ phải gán bit 1. Thay thế mỗi phần tử bằng mã tiền tố tương ứng cho đến hết chuỗi. Ghi ra file nhị phân từng block 8 bits.

Lưu đồ thuật toán:



Chạy thử chương trình:

1. Nén file chuỗi (.txt)

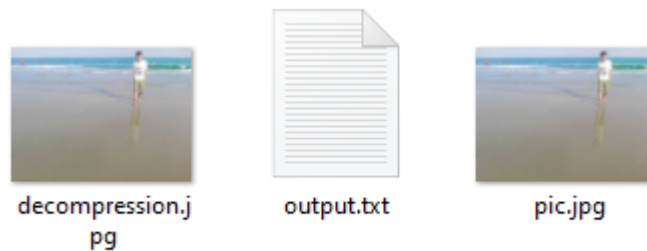
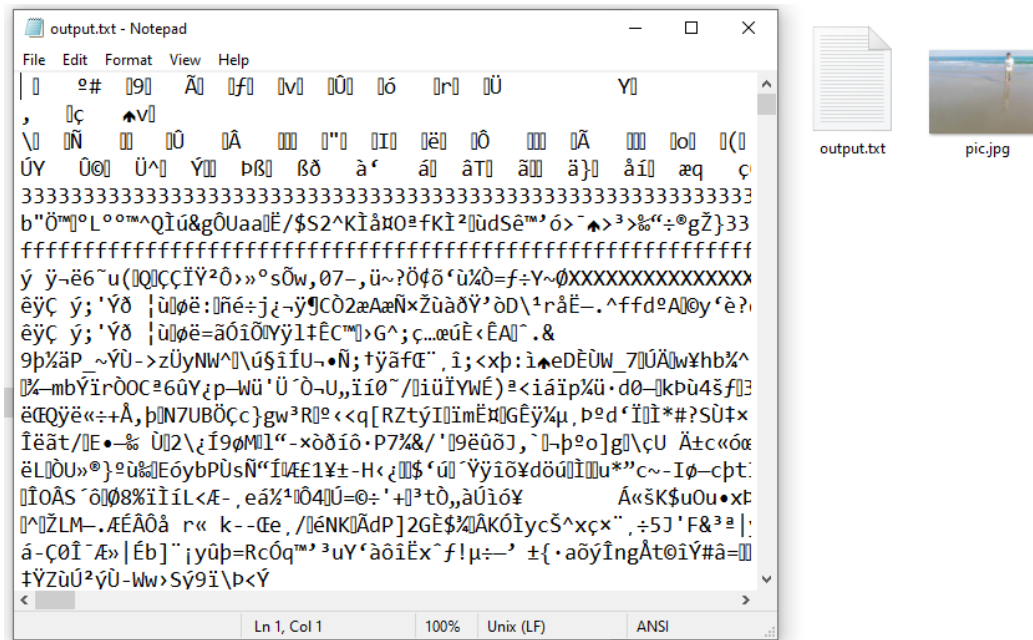


Name	Date modified	Type	Size
decompression.txt	13/12/2019 5:45 PM	Text Document	363 KB
input.txt	04/12/2019 7:22 PM	Text Document	363 KB
output.txt	13/12/2019 5:45 PM	Text Document	246 KB

file **decompression.txt** sau khi giải nén có size = 363KB bằng size của file input, như vậy thao tác nén thành công

file **output.txt** sau khi nén có size 246KB < 363KB của file input.txt trước khi nén

2. Nén file ảnh (.jpg)



3. Nén file .cpp

++ Compression.cpp	07/12/2019 3:54 PM	C++ Source	9 KB
++ decompression.cpp	13/12/2019 6:00 PM	C++ Source	9 KB
output.txt	13/12/2019 6:00 PM	Text Document	6 KB

4. Test trên file dữ liệu mẫu

file output.txt đã được nén từ file corpus-title.txt có size nhỏ hơn file gốc

corpus-title.txt	12/02/2018 11:20 PM	Text Document	591,449 KB
output.txt	14/12/2019 11:44 AM	Text Document	398,749 KB

file decompression.txt được giải nén từ file output.txt có size bằng file gốc

corpus-title.txt	12/02/2018 11:20 PM	Text Document	591,449 KB
decompression.txt	14/12/2019 11:49 AM	Text Document	591,449 KB

3 Tham khảo & Video Demo

Video Demo:

<https://www.youtube.com/watch?v=2vbQsF3a5PY>

Link github:

https://github.com/ThienAn212/CTDLvaGT?fbclid=IwAR1aVRKWimQ4UTpjJTfoHy8nnX2c6X-kaKJrskhOJ7Ra_7UQDqXVuENvDfc

Tài liệu tham khảo:

<http://diendan.congdongcviet.com/threads/t2381::thuat-toan-nen-du-lieu-ly-thuyet-nen-du-lieu-huffman-phan-1-y-tuong.cpp>