BẢN BÁO CÁO KỸ THUẬT

Họ tên: Hoàng Văn Thức

Mssv:1653139

1. Cấu trúc file nén

-Mã nhận dạng file kiểu char 8 ký tự

-Bảng tần số (unsigned int):

-Số lượng file : kiểu unsigned int

-Kế tiếp là field chứa những thông tin của file, có bao nhiêu file thì có bấy nhiêu field gồm:

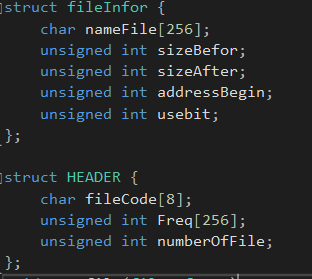
+Tên file gồm 256 ký tự kiểu char

+size của file trước khi nén

+size của file sau khi nén

+địa chỉ bắt đầu của thông tin sau khi nén của file đó

+số bit cần dùng



1. Các cấu trúc dữ liệu khác của chương trình:

* Hàng đợi ưu tiên: Dùng để xây dựng cây Huffman: Đưa tất cả các node lá của cây Huffman vào hàng đợi ưu tiên,sau đó tạ node cha từ 2 node con đã được dequeue, sau đó enqueue node cha đó lại vào hàng đợi ưu tiên. Lặp cho đến khi hàng đợi rỗng.

1. Các hàm chính của chương trình:

+ void findFilesInFolder(const string input, string a[], unsigned int sizefile[]);

Đọc lần lượt các file trong folder mà khách hàng nhập vào và lấy thông tin của file gồm tên của file và kích thước của file trước khi nén. (Sử dụng findfirstfile và filenextfile trong window.h). Đồng thời gọi hàm makeFrequencyStatistics để tạo bảng tần số (mỗi lần đọc 2 sector cho vào mảng 1024 phần tử kiểu char)

-input: đường dẫn của folder

-output: trả về mảng tên các file và mảng kích thước các file trong thư mục

+void makeFrequencyStatistics(unsigned char buff[1024], unsigned char stopIndex);

Đọc từng kí tự trong mảng 1024 kí tự và lập bảng tần số.

-input: mảng buff 1024 phần tử, và chỉ số dừng, vì khi cuối file chưa chắc sẽ đủ 1024 kí tự để cho vào mảng.

-output: bảng thống kê tần số (được khai báo toàn cục)

+ void moveFreqToQueue(PRIORITY\_QUEUE &a);

Đưa tất cả node lá vào hàng đợi ưu tiên để xử lý khi tạo cây huffman.

-input: Hàng đợi ưu tiên kiểu tham biến, bảng tần số ( được khai báo toàn cục).

-output: Hàng đợi ưu tiên đã được xây dựng dựa trên bảng tần số.

+int makeHuffmanTree(PRIORITY\_QUEUE &a);

Tạo cây huffman trên mảng gồm 511 node, dequeue 2 lần để lấy giá trị và tạo 1 node cha dựa trên node con, sau đó enqueue node cha vào hàng đợi ưu tiên.Lặp lại tới khi hàng đợi ưu tiên rỗng .

-input: Hàng đợi ưu tiên đã được xây dựng từ trước

-output: Mảng gồm 511 phần tử ( đã được khai báo toàn cục) đã chứa thông tin của cây huffman

+void traverseHuffmanTree(int node, char bitcode[], int nbit) và void makeBitCode(int nRoot);

Duyêt cây huffman. Đi qua trái thì là cho 0 vào mảng bitcode, đi qua phải thì thêm 1 vào mảng bitcode, dùng đệ quy, duyệt đến khi nLeft, nRight đều bằng -1 thì dừng.

Sau đó cho vào bảng mã bit đã được khai báo toàn cục.

-input: node gốc, mảng mã bit, số lượng bit

-output: tạo ra bảng bitcode.

+void encodeAChar(unsigned char c, unsigned char &out, unsigned char &bitIndex, ofstream &fileout);

Chuyển 1 kí tự thành mã bit, cho từng bit vào 1 byte, đủ 8 bit thì in ra file

-input: Kí tự cần mã hóa, kí tự chứa bit, vị trí của bit m và ofstream &fileout để in ra

-output: In ra kí tự out vào file

+void compressedFile(string inputFile);

Là hàm để nén file, đọc tất cả các file, lấy từng kí tự , mã hóa từng kí tự viết vào file.

-input: Đường dẫn folder

-output: File.huff (nhị phân) chứa cấu trúc file ở trên và thông tin nén

+void decodeAChar(unsigned char c, char bitOfChar[8])

Chuyển 1 kí tự thành mã bit để duyệt cây

-input: kí tự kiểu char c, mảng bitOfChar

-output-: Mảng char 8 kí tự chứa thông tin bit của kí tự char c

+void makeCharFromBit(int node, char bitcode[], char nbit, ofstream &fileout)

Từ mảng chứa các bit được lấy từ hàm decodeAchar đưa vào hàm này để chuyển thành kí tự và viết vào file, nếu là 0 thì đi qua trái của cây, là 1 thì đi qua phải .

-Input: node bắt đầu duyệt, mảng bitcode, chỉ số nbit, fileout

-output: kí tự hoàn chỉnh và viết vào fileout