|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN KHOA CÔNG NGHỆ THÔNG TIN** |  |

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT  
BÁO CÁO ĐỒ ÁN 2 – ỨNG DỤNG NÉN DỮ LIỆU**

GIÁO VIÊN HƯỚNG DẪN: THẦY NGUYỄN THÀNH AN

HỌ VÀ TÊN: LÊ NHẬT TUẤN – MSSV: 18120632

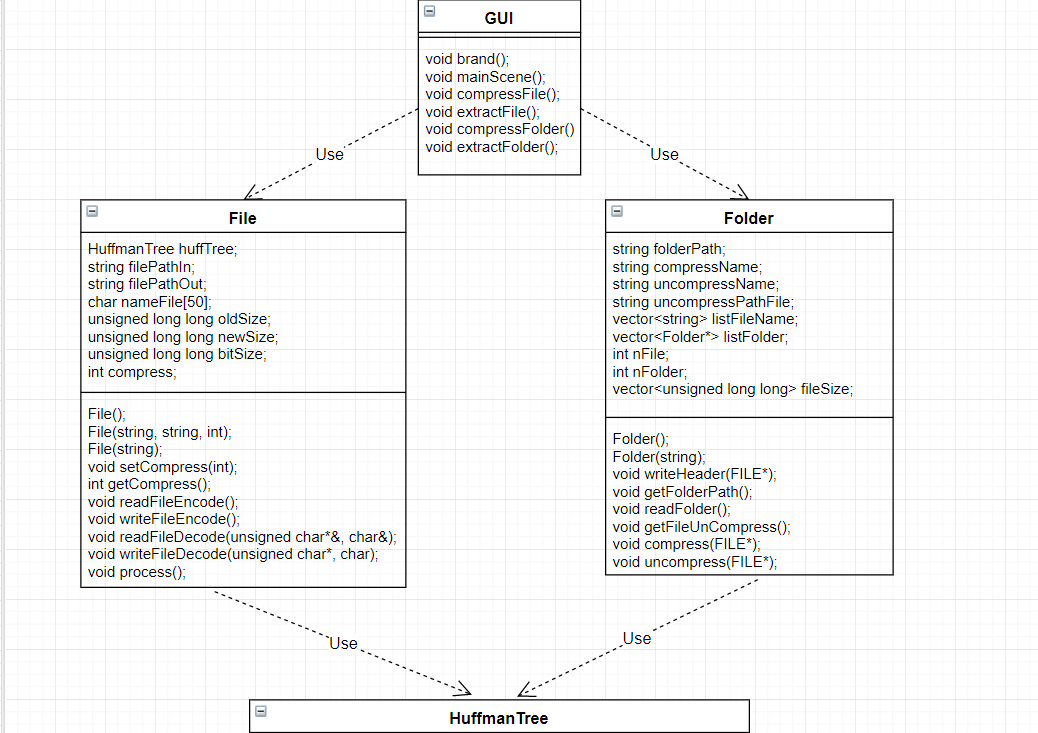
1. **Câu làm được:**

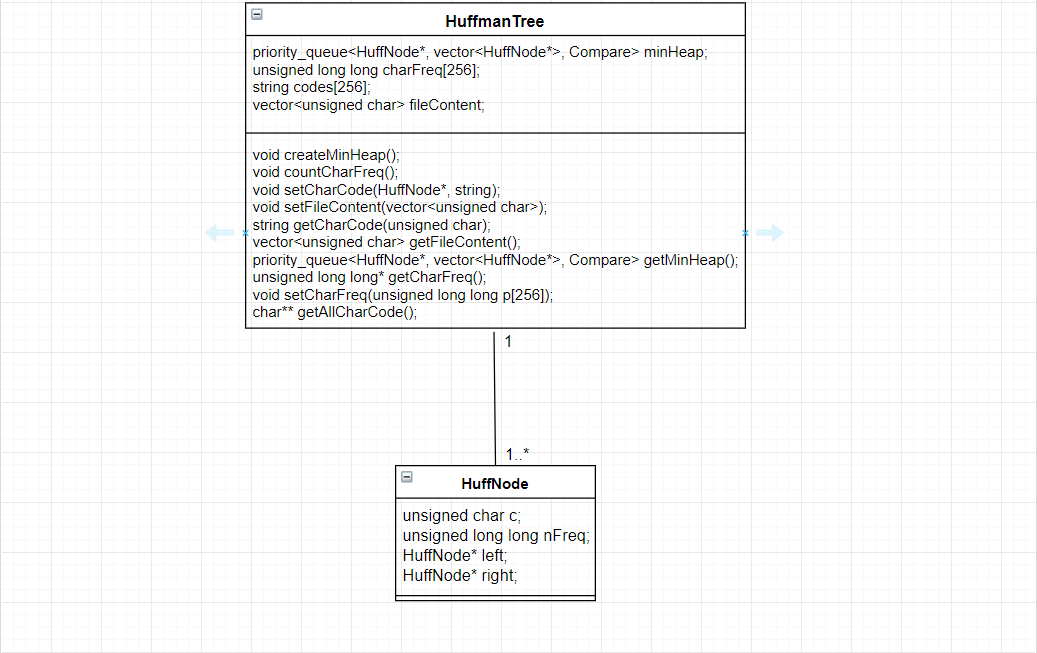
* Nén 1 tập tin chuỗi.
* Nén 1 thư mục chứa nhiều tập tin chuỗi.
* Nén 1 tập tin chưa những kiểu dữ liệu còn lại.
* Nén 1 thư mục chưa nhiều tập tin với những kiểu dữ liệu còn lại.

1. **Câu chưa làm được:**

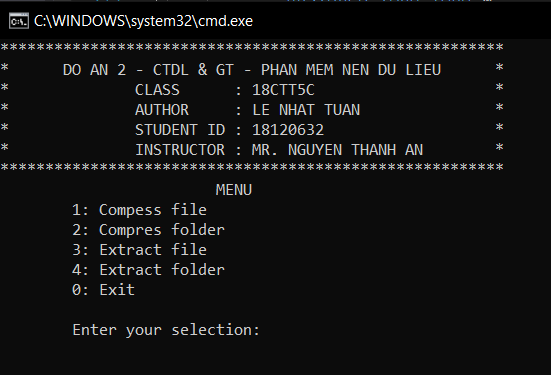
* Không có.

1. **Sơ đồ lớp:**



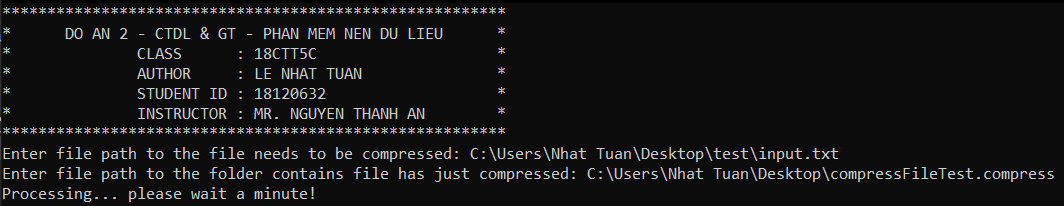


1. **Hướng dẫn sử dụng:**
2. Giao diện bắt đầu:

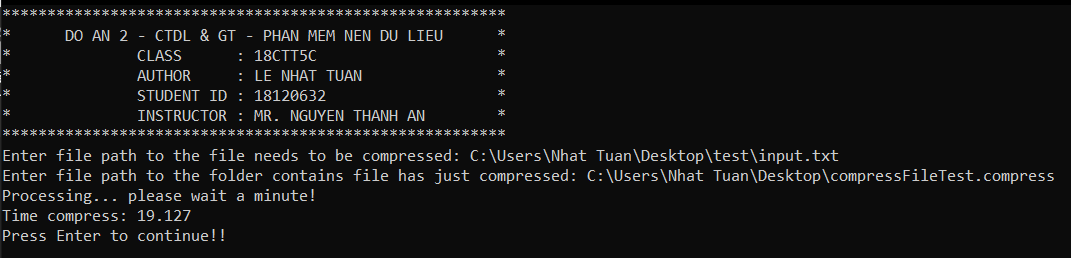


1. Chọn 1: Nén file.

Nhập vào đường dẫn đến file cần nén, nhập vào đường dẫn đến nơi chưa file nén và tên file nén:

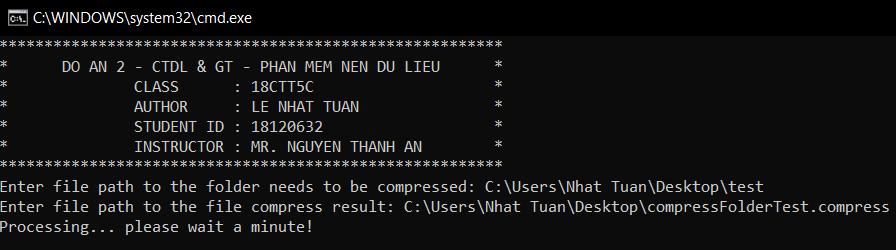


Đang xử lí và kết quả:

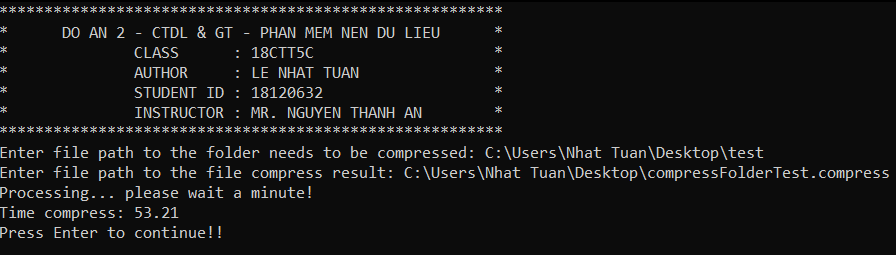


1. Chọn 2: Nén folder

Nhập vào đường dẫn đến folder cần nén, nhập vào đường dẫn đến nơi chứa file nén và tên file nén

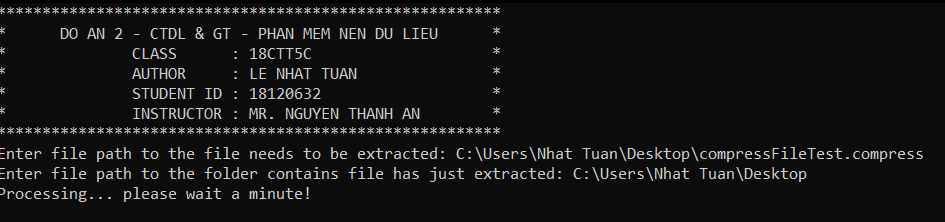


Đang xử lí và kết quả

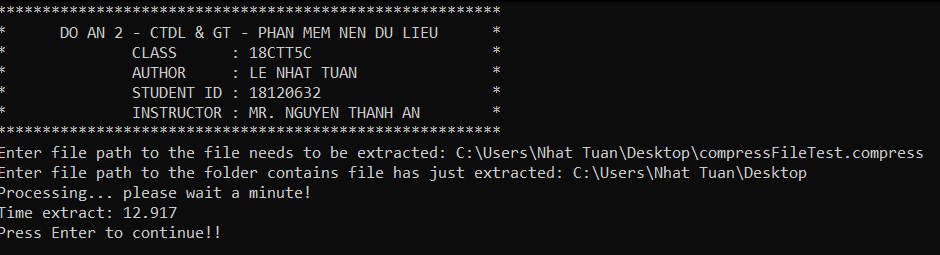


1. Chọn 3: Giải nén file

Nhập vào đường dẫn đến file cần giải nén, nhập vào đường dẫn đến nơi chứa file giải nén:

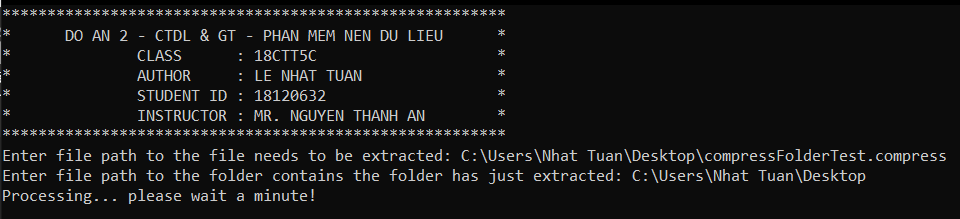


Đang xử lí và kết quả:

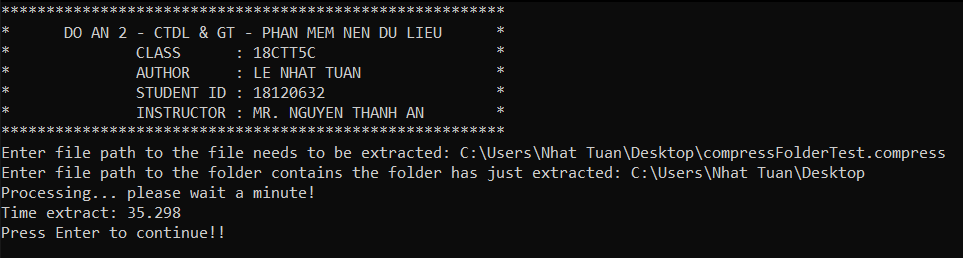


1. Chọn 4: Giải nén folder

Nhập vào đường dẫn đến file cần giải nén, nhập vào đường dẫn đến nơi chứa file giải nén:



Đang xử lí và kết quả:



1. **Giải thích thuật toán nén dữ liệu sử dụng: Thuật toán nén HUFFMAN**
2. **Ý tưởng:**
   * Sử dụng số lượng bit khác 8 bit để biểu diễn 1 kí tự trong bảng mã ASCII.
3. **Xây dựng cây HUFFMAN:**

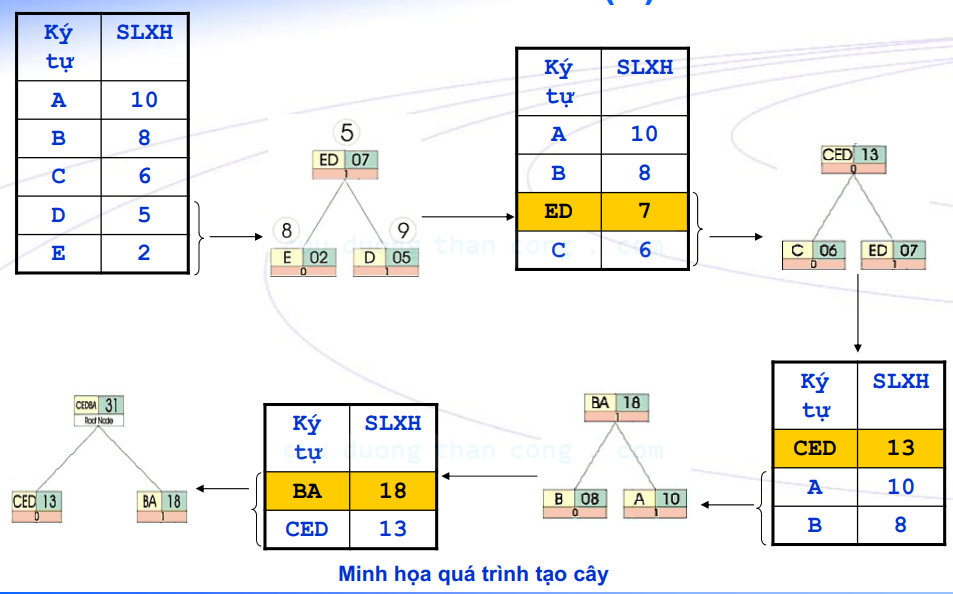
**Bước 1:** Đọc theo kiểu nhị phân toàn bộ file cần nén. Mỗi lần đọc 1 byte cho đến khi hết file. Vừa đọc vừa đồng thời tăng số lần xuất hiện của kí tự vừa đọc được trong bảng tần số.

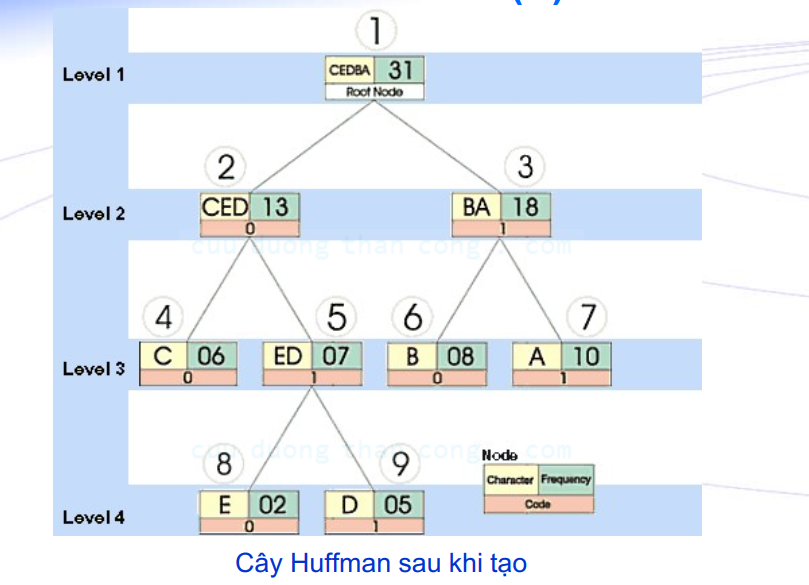
**Bước 2:** Từ bảng tần số tạo ra các NODE với các thành phần: Con trỏ trái, con trỏ phải, kí tự, số lần xuất hiện. Thêm tất cả các NODE vừa tạo vào 1 bảng.

**Bước 3:** Ghép 2 NODE có số lần xuất hiện ít nhất thành NODE mới với số lần xuất hiện bằng tổng số lần xuất hiện của 2 NODE vừa được ghép. Con trỏ trái trỏ đến NODE có số lần xuất hiện ít hơn. Con trỏ phải đến NODE còn lại. Trong trường hợp 2 NODE có số lần xuất hiện bằng nhau thì con trỏ trái của NODE mới trỏ dến NODE có kí tự có mã ascii nhỏ hơn. Con trỏ phải trỏ đến NODE còn lại. Kí tự của NODE mới được gán bằng kí tự của NODE mà con trỏ trái nó trỏ đến.

**Bước 4:** Loại bỏ 2 NODE vừa được ghép và thêm NODE mới vào bảng.

**Bước 5:** Nếu bảng chỉ còn 1 NODE thì dừng lại. Ngược lại quay lại Bước 3:





1. **Phát sinh mã CODE cho từng kí tự:**
   * Quy luật:
     + Duyệt sang trái: cộng thêm ‘0’ vào mã code.
     + Duyệt sang phải: cộng thêm ‘1’ vài mã code.
   * Thực hiện: sử dụng đệ quy.
     + Đặt mã code ban đầu bằng “” (Rỗng).
     + Duyệt tuần từ từ trên xuống dưới, trái sang phải. Mỗi lần duyệt thì cộng ‘0’ hoặc ‘1’ theo quy luật duyệt tương ứng.
     + Duyệt cho đến khi gặp nút lá thì gán kí tự ở nút lá bằng mã code vừa sinh được.
     + Duyệt đến khi nào hết tất cả các nút lá trong cây thì dừng lại
2. **Tạo file nén:**
   * Ghi bảng tần số số lần suất hiện của từng kí tự trong file cần nén lên file nén.
   * Đọc lại file cần nén. Mỗi lần đọc 1 byte. Vừa đọc vừa thay thế kí tự đó bằng mã code tương ứng. Thêm mã code đó vào chuỗi kết quả.
   * Sau khi tạo được chuỗi kết quả. Thêm các kí tự 0 cho chuỗi kết quả. Sao cho chuỗi kết quả có độ dài chia hết cho 8. Ghi phần padding là số kí tự ‘0’ được thêm vào lên file nén.
   * Tách lần lượt 8 kí tự trong chuỗi kết quả tương ứng với 8 bit và phát sinh ra số ở hệ thập phân tương ứng và ghi số đó lên file nén. Thực hiện cho đến khi nào đọc hết chuỗi mã kết quả.
3. **Giải nén:**

**Bước 1:** Đọc bảng tần số được ghi trên file nén.

**Bước 2:** Dựa vào bảng tần số xây dựng cây HUFFMAN mới và padding.

**Bước 3:** Đọc các toàn bộ file nén, mỗi lần đọc 1 byte.

**Bước 4:** Đổi các byte vừa đọc thành mã nhị phân rồi thêm vào chuỗi cần giải nén.

**Bước 5:** Từ chuỗi cần giải và cây HUFFMAN vùa tạo được.Đặt NODE tạm là NODE root của cây HUFFMAN. Duyệt lần lượt từ đầu chuỗi đến cuối chuỗi. Gặp ‘0’ thì NODE tạm duyệt sang trái, ‘1’ thì duyệt NODE tạm sang phải, duyệt đến khi nào NODE tạm trở thành NODE lá thì ghi kí tự mà NODE lá chứa lên file giải nén, đồng thời đưa NODE tạm trở về lại vị trí NODE root.

**Bước 6:** Ở byte cuối cùng. Chỉ duyệt đến số lượng bit bằng 8 – padding.

1. **Link youtube video hướng dẫn:**
2. **Link GitHub:**