Để nén dữ liệu, ta sử dụng thuật toán Huffman:

1. Biểu diễn dữ liệu:

Thông thường, ta dùng 8 bit để mã hóa 1 ký tự, ví dụ như:
“a” = 01100001
“A” = 01000001
Vậy, với bộ mã ASCII, có thể mã hóa 2^8 = 256 ký tự, với mỗi từ mã dài 8 bit.

2. Vấn đề phát sinh:

Ví dụ, để lưu trữ file text có nội dung: “ABABAC”
Nếu dùng 8 bit để biểu diễn 1 ký tự thì tiêu tốn 6*8 = 48 bits

3. Ý tưởng:

Ta thấy tần suất xuất hiện các ký tự trong dữ liệu:
A: 3 lần
B: 2 lần
C: 1 lần
Để ý trong nội dung tập tin chứa các ký tự trùng nhau, bị lặp lại nhiều lần. Hơn nữa, ta có thể không cần dùng đủ 8 bít để mã hóa cho một ký tự. Cần lập bảng mã sao cho độ dài (số bit) dành cho mỗi kí hiệu có thể khác nhau, kí hiệu nào xuất hiện nhiều lần thì nên dùng số bít ít, ký hiệu nào xuất hiện ít thì có thể mã hóa bằng từ mã dài hơn. Tuy nhiên, nếu mã hóa với độ dài thay đổi, thì khi giải mã ta làm thế nào phân biệt được xâu bít nào là mã hóa của ký hiệu nào? Và mã tiền tố là giải pháp phù hợp trong trường hợp này.

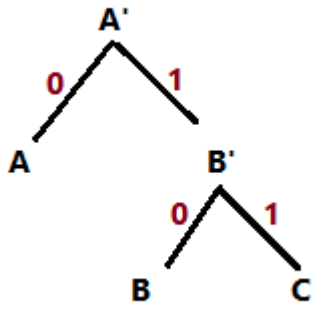
4. Thuật toán xây dựng mã tiền tố:

- Bước 0: Khởi tạo một rừng n cây, mỗi cây chỉ có 1 nút gốc và 1 nút lá. Nút lá chứa ký tự, và nút gốc chứa tần số xuất hiện của ký tự đó.

- Bước 1: Chọn 2 cây có trọng số ở gốc nhỏ nhất hợp thành một cây bằng cách thêm một gốc mới nối với hai gốc đã chọn. Trọng số của gốc mới bằng tổng trộng số của hai gốc tạo thành nó. (Ở đây, trọng số chính là tần số xuất hiện của ký tự đó trong tập dữ liệu).

- Bước 2: Lặp lại quá trình này đến khi rừng chỉ còn một cây.

Sau khi xây dựng cây nhị phân tiền tố, ta duyệt cây để tìm ra mã tiền tố cho mỗi ký tự với giả thuyết rẽ nhánh trái mang bit 0, rẽ nhánh phải mang bit 1. Ví dụ:



Từ đó, chúng ta tự lập ra 1 bảng mã tiền tố như sau:
A: 0
B: 10
C: 11
Như vậy, ta tiết kiệm được 48 – (3*1+2*2+1*2) = 39 bits

Chuỗi dữ liệu sử dụng mã tiền tố được biểu diễn dạng nhị phân: 010010011

5. Giải thuật nén file:

Giả sử đã xây dựng được bộ mã Huffman cho tất cả các ký tự, chúng ta có thể nén file theo giải thuật sau:
- Đọc từng byte của file cần nén cho đến khi hết tệp
- Chuyển theo bộ mã thành xâu nhị phân
- Ghép với xâu nhị phân còn dư từ bước trước
- Nếu có đủ 8 bit trong xâu thì cắt ra 8 bít đầu ghi vào tệp nén
- Nếu đã hết tệp cần nén thì dừng

Ví dụ, theo gia thuyết ở trên, ta có xâu nhị phân: 010010011

Xâu này được tách ra từng bộ 8 bits từ trái sang phải: 01001001 | 1

Ghi từng block 8 bits lên file nhị phân, ta được file nén có size nhỏ hơn file gốc.