Sơ đồ nén MP3

1. ***Số hóa bằng cách lấy mẫu theo khoảng.*** Tạo ra một dãy số thực

$$s_1, s_2, \ldots, s_T$$

Ví dụ, với tốc độ $44,100$ mẫu trên giây, bản giao hưởng 50 phút có

$$T = 50 \times 60 \times 44,100 \approx 130 \text{ triệu.}$$

2. ***Lượng hóa.*** Xấp xỉ s_i bởi giá trị gần nhất thuộc tập hữu hạn Γ.
3. ***Mã hóa.*** Xâu $s_1 s_2 \ldots s_T$ trên bảng chữ Γ được mã hóa ở dạng nhị phân. (Dùng mã Huffman)

Mã hóa

Ký hiệu	Số lần xuất hiện
A	70 triệu
B	3 triệu
C	20 triệu
D	37 triệu

- Bảng chữ $\Gamma = \{A, B, C, D\}$ và $T = 130$ triệu.
- Nếu mã hóa dùng 2 bit cho mỗi ký hiệu, ví dụ
$$A \to 00, \qquad B \to 01, \qquad C \to 10, \qquad D \to 11$$
ta cần 260 megabits.
- Liệu ta có thể dùng ***mã độ dài thay đổi*** để giảm kích thước bản mã?

Mã độ dài thay đổi

- Dùng các dãy bit độ dài khác nhau để mã hóa các chữ cái.
- Chữ cái xuất hiện thường xuyên hơn sẽ được mã bằng dãy bit ngắn hơn.
- Vấn đề: Làm thế nào xác định được mỗi chữ bắt đầu và kết thúc ở đâu trong dãy bit.?

Ví dụ

Cách mã hóa

$$A \to 0, \qquad C \to 01, \qquad D \to 11, \qquad B \to 001$$

gây ra nhập nhằng khi giải mã

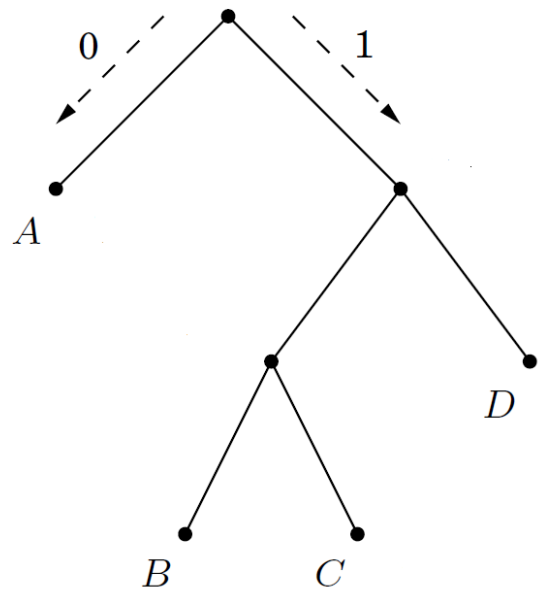
$$001 \to AC$$
$$001 \to B$$

Mã tiền tố

Định nghĩa

Mã tiền tố là tập xâu thỏa mãn không có xâu nào là khúc đầu của xâu khác.

Symbol	Codeword
A	0
B	100
C	101
D	11



Hãy giải mã dãy bit 10100100?

Kích thước bản mã

Ký hiệu	Số lần xuất hiện	Từ mã
A	70 triệu	0
B	3 triệu	100
C	20 triệu	101
D	37 triệu	11

- Kích thước bản mã

$$= (1 \times 70 + 3 \times 3 + 3 \times 20 + 2 \times 37) \text{ megabits}$$
$$= 213 \text{ megabits}$$

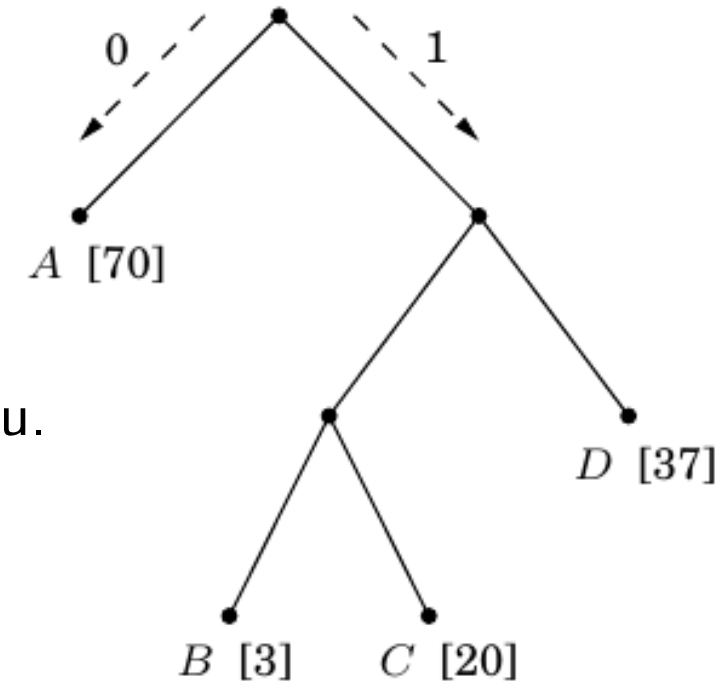
- Cải thiện 17% so với 260 megabits khi dùng mã độ dài cố định.

Bài toán

- Cho n ký hiệu có tần suất

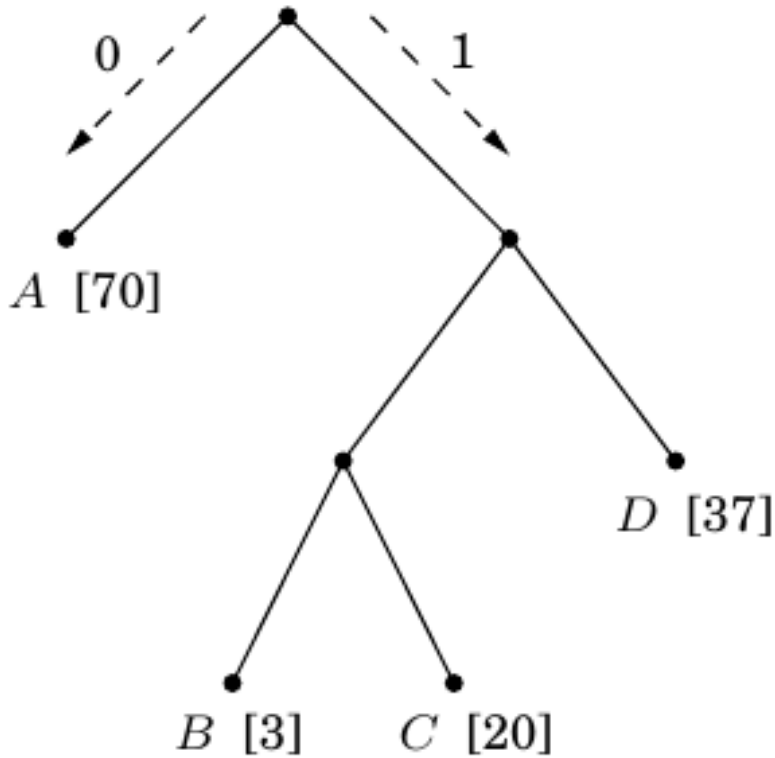
$$f_1, f_2, \ldots, f_n.$$

- Hãy tìm cây ở đó mỗi lá ứng với một ký hiệu và có chi phí cực tiểu.



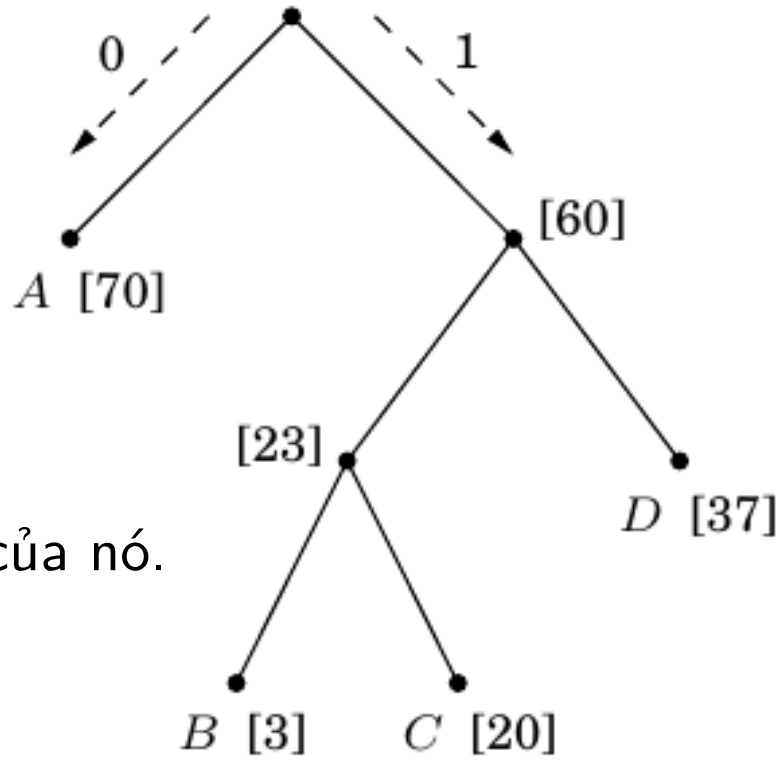
$$\text{Chi phí của cây } = \sum_{i=1}^{n} f_i \cdot (\text{độ sâu ký hiệu thứ } i \text{ trong cây})$$

Hãy tính chi phí của cây sau.



$$\text{Chi phí của cây } = \sum_{i=1}^{n} f_i \cdot (\text{độ sâu ký hiệu thứ } i \text{ trong cây})$$

- Tần suất nút ***lá*** là f_i.
- Tần suất ***nút trong*** là tổng tần suất của các ***lá*** con cháu của nó.



Mệnh đề

Chi phí của cây là tổng tần suất của mọi nút ngoại trừ nút gốc.

Tối ưu hàm chi phí

$$\text{Chi phí của cây } = \sum_{i=1}^{n} f_i \cdot (\text{độ sâu của ký hiệu thứ } i \text{ trong cây})$$

Nhận xét

Hai ký hiệu với tần suất nhỏ nhất sẽ phải ở đáy của cây tối ưu.

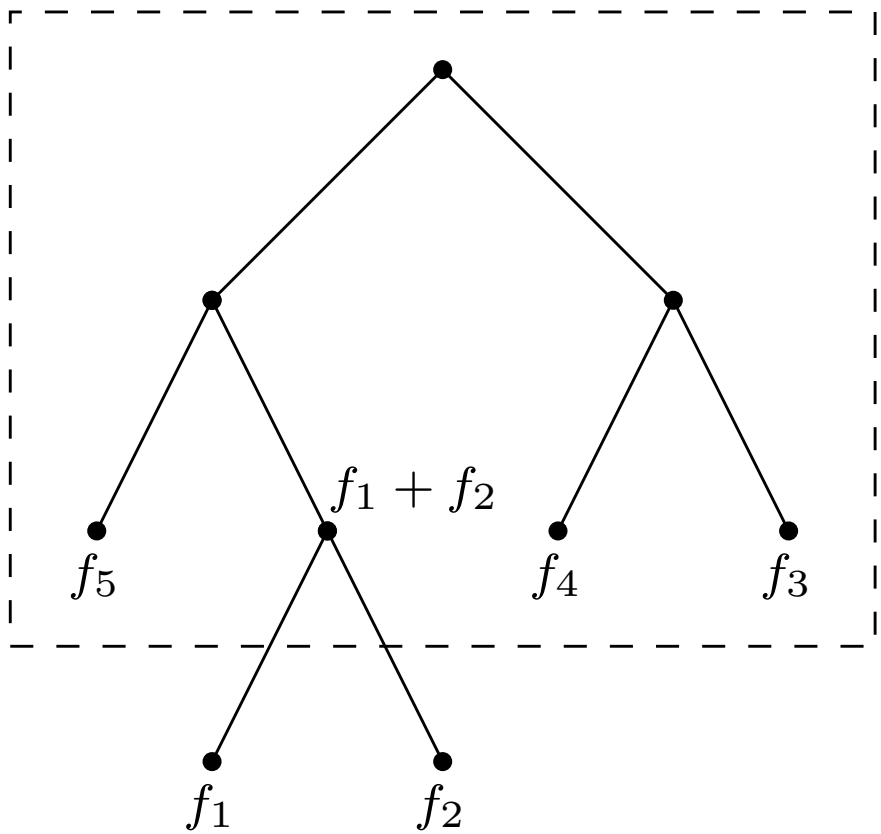
Xây dựng cây một cách tham lam

- Tìm hai ký hiệu có tần suất nhỏ nhất, gọi là i và j, và tạo nút cha của chúng với tần suất $f_i + f_j$.
  Để đơn giản ký hiệu, ta giả sử chúng là f_1 và f_2.
- Mọi cây trong đó f_1 và f_2 là nút lá anh em có chi phí $f_1 + f_2$ cộng với chi phí cho cây gồm $n - 1$ nút lá của các tần suất:

$$(f_1 + f_2),\ f_3,\ f_4, \ldots, f_n.$$

- Ta đưa về bài toán kích thước nhỏ hơn. Ta loại bỏ f_1 và f_2 khỏi dãy tần suất và thêm $(f_1 + f_2)$ vào, và ***lặp lại***.

Xây dựng cây một cách tham lam



Hình: Loại f_1, f_2 và thêm $f_1 + f_2$ vào dãy tần suất.

procedure Huffman(f)
Input: mảng $f[1 \cdots n]$ của các tần suất
Output: Một cây mã hóa với n lá

Xét H là hàng đợi ưu tiên của các số nguyên, thứ tự bởi f
for $i = 1$ to n: insert(H, i)
for $k = n + 1$ to $2n - 1$:
    i = deletemin(H), j = deletemin(H)
    Tạo một nút đánh số k với các con là i, j
    $f[k] = f[i] + f[j]$
    insert(H, k)