



Trường Đại Học Công Nghệ Thông Tin

CS232 - Tính toán đa phương tiện

Data Compression with Huffman, LZW

Lớp:

CS232.M21.KHCL

Giảng viên hướng dẫn:

TS. Đỗ Văn Tiến

Các thành viên trong nhóm



► Gồm các thành viên:

MSSV	Họ & Tên
19522274	Hồ Thịnh
19522093	Lê Vinh Quang
19522432	Nguyễn Thành Trung



- I. Nén dữ liệu
- II. Huffman Coding (Static Huffman)
- III. Lempel-Ziv Welch (LZW)
- IV. Kết quả thực nghiệm
- V. Demo



I. Nén dữ liệu

I. Nén dữ liệu



- ▶ Nén dữ liệu là quá trình mã hóa thông tin dùng ít bit hơn so với thông tin chưa được mã hóa bằng cách dùng một hay kết hợp của các phương pháp nào đó.
- ▶ Dựa theo nguyên tắc này giúp tránh các hiện tượng kênh truyền bị quá tải và việc truyền tin trở nên hiệu quả hơn. Nén dữ liệu giúp tiết kiệm các tài nguyên như dung lượng bộ nhớ, băng thông, thời gian. Ngược lại, dữ liệu đã được nén cần được giải nén để đọc (thực thi, nghe, xem v.v...), quá trình này cũng đòi hỏi các tài nguyên nhất định.

Dẫn nhập: Mã hoá dữ liệu trong máy tính



“ARRAY”



Character	ASCII
A	100 0001
R	101 0010
Y	101 1001



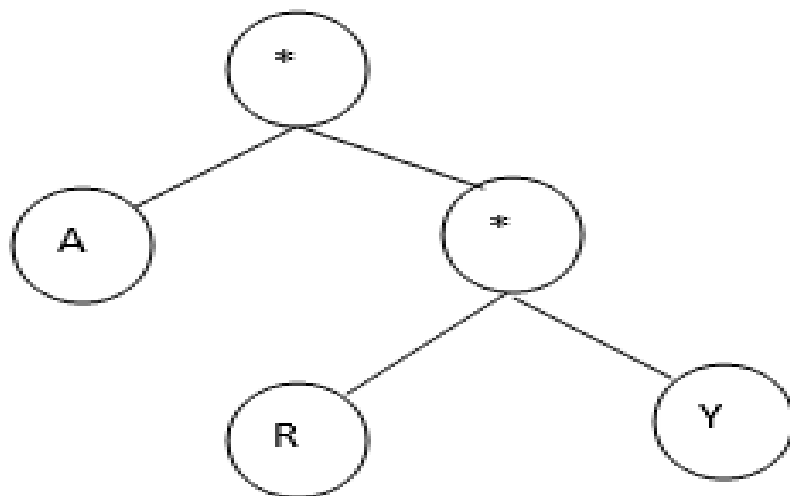
10000011010010101001010000011011001

Có thể gây lãng phí khi phải biểu diễn cho những kí tự xuất hiện ít và chúng ta vẫn có thể không sử dụng hết 256 ký hiệu khi mã hoá một tài liệu

Dẫn nhập: Mã hoá dữ liệu trong máy tính



"ARRAY"



Char	Code
A	0
R	10
Y	11

II. Huffman Coding (Static Huffman)



- ▶ Ý tưởng: Ký tự xuất hiện càng nhiều thì gán mã càng ngắn và ngược lại
- ▶ Các bước thực hiện:
 - Bước 1: Duyệt dữ liệu để lập bảng thống kê số lần xuất hiện của mỗi ký tự
 - Bước 2: Tạo cây Huffman từ bảng thống kê
 - Bước 3: Tạo bảng mã cho các ký tự
 - Bước 4: Duyệt dữ liệu để thay thế các ký tự bằng mã bit tương ứng
 - Bước 5: Lưu lại thông tin của cây Huffman dùng để giải nén

II.1. Cây Huffman



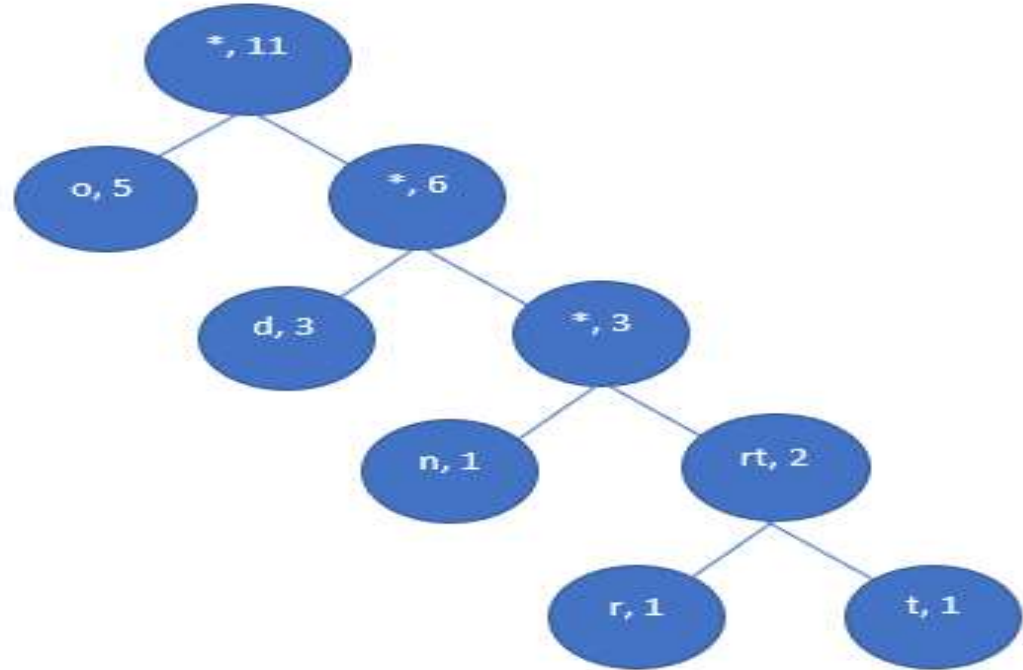
- ▶ Cây Huffman là binary tree với giá trị 0 thể hiện cho nhánh trái và 1 cho nhánh phải.
- ▶ Những giá trị trên nút của cây đại diện cho tổng tần số (tổng giá trị của các nút con).
- ▶ Nút lá của cây đại diện cho các ký tự có tần suất xuất hiện ít nhất.

II.2. Ví dụ minh họa



“do or do not do”

Char	Freq
o	5
d	3
n	1
r	1
t	1



II.3. Một số nhược điểm



- ❑ Duyệt dữ liệu 2 lần (thống kê và mã hoá) chi phí tính toán cao
- ❑ Phải lưu trữ cây Huffman tăng kích thước dữ liệu nén
- ❑ Dữ liệu nén phải có sẵn đầy đủ, không nén được với các dữ liệu phát sinh theo thời gian thực

II.4. Adaptive Huffman



- ❑ Được đề xuất bởi Faller (1973) và Gallager (1978). I Knuth (1985) đưa ra một số cải tiến và hoàn chỉnh thuật toán và thuật toán còn có tên “thuật toán FGK”
- ❑ Các cải tiến:
 - Không cần tính trước số lần xuất hiện của các ký tự
 - Quá trình nén chỉ cần 1 lần duyệt dữ liệu
 - Không cần lưu thông tin phục vụ cho việc giải nén
 - Nén “online” dựa trên dữ liệu phát sinh theo thời gian thực



III. Lempel-Ziv Welch

III. Lempel-Ziv Welch



- ▶ **Ý tưởng:** Xây dựng một bảng mã, bảng mã này không được lưu kèm với dữ liệu trong quá trình nén, và khi giải nén bộ giải nén sẽ xây dựng lại nó.
- ▶ **Nguyên tắc hoạt động:**
 - Một chuỗi ký tự là một tập hợp gồm 2 ký tự trở lên
 - Có một từ điển để lưu các chuỗi ký tự đã duyệt và gán chúng bằng các token riêng
 - Nếu duyệt lại chuỗi ký tự đã có trong từ điển thì gán lại token

III.1. Các quy tắc thực hiện nén



- ▶ 256 dấu hiệu đầu tiên được dành cho các ký tự đơn
- ▶ Cố gắng so sánh với từ điển khi trong bộ đệm chứa đã có nhiều hơn hai ký tự
- ▶ Các ký tự ở input được bổ sung vào bộ đệm chứa đến khi chuỗi ký tự trong bộ đệm chứa không có trong từ điển
- ▶ Khi bộ đệm chứa có một chuỗi mà trong từ điển không có thì chuỗi trong bộ đệm chứa được đem vào từ điển. Ký tự cuối cùng của chuỗi ký tự trong bộ đệm chứa phải ở lại trong bộ đệm chứa để tiếp tục tạo thành chuỗi mới

III.2. Ví dụ minh họa



“doordonot”

$P=t$

$C = \text{empty}$

Encoder	Output	String	Table
Output Code	representing	codeword	string
100	d	256	do
111	o	257	oo
111	o	258	or
114	r	259	rd
256	do	260	don
110	n	261	no
111	o	262	ot

Bước 7

III.3. Ưu điểm so với Huffman Coding



- LZW không yêu cầu thông tin trước về luồng dữ liệu đầu vào.
- LZW có thể nén luồng đầu vào trong một lần truyền.
- Một ưu điểm khác của LZW là tính đơn giản, cho phép thực thi nhanh chóng



IV. Kết quả thực nghiệm

IV.1. Nén ảnh màu sử dụng Huffman Coding



image.png
Size: 4KB
Original size: 708479KB

imageCompressed.txt
Size: 2776KB
Compressed size: 355248KB

Compression Ratio - 2:1

IV.1. Nén ảnh màu/gif sử dụng LZW



image.png
Size: 4KB
Original size: 151008KB

imageCompressed.lzw
Size: 2776KB
Compressed size: 43463KB

Compression Ratio -



V. Demo



**Thank you for
Watching**