# XỬ LÝ ẢNH CÁC PHƯƠNG PHÁP NÉN ẢNH

Giảng Viên: ThS. Đinh Phú Hùng

Bộ môn: Khoa Học Máy Tính

Email: hungdp@tlu.edu.vn

03/2021

#### Nội Dung

- Tổng quan về nén ảnh
- 2 Phân loại các phương pháp nén ảnh
- 3 Một số phương pháp nén ảnh

# Tổng quan về nén ảnh

#### Môt số khái niêm cơ bản:

- Dư thừa dữ liệu và thông tin.
- Nén dữ liêu.
- Tỉ lê nén.
- Tính dung lượng ảnh.

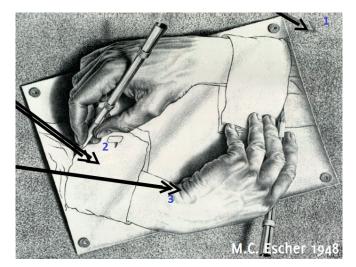
## Dư thừa dữ liệu và thông tin

- Dữ liệu và thông tin không tương đương với nhau.
- Dữ liệu là phương tiện để thể hiện thông tin. Số lượng dữ liệu có thể nhiều hơn số lượng thông tin.
- Những dữ liệu dư thừa không cung cấp thêm thông tin.
- Nén ảnh mục đích là để giảm bớt lượng dữ liệu dư thừa trong khi vẫn giữ nguyên được lượng thông tin.

# Một số kiểu dư thừa dữ liệu trong ảnh

- Kiểu 1: Dư thừa mã
   Một số mức xám trong ảnh phổ biến hơn các mức xám còn lại.
- Kiểu 2: Dư thừa trong pixel
   Giá trị của các pixel trong ảnh có thể được dự đoán hợp lý từ các pixel láng giềng.
- Kiểu 3: Dư thừa trong thị giác mắt
   Mắt người chỉ có thể phân biệt được một số lượng màu sắc nhất định.

# Một số kiểu dư thừa dữ liệu trong ảnh



#### Nén dữ liêu

- Nén dữ liệu nhằm làm giảm lượng thông tin "dư thừa" trong dữ liệu gốc.
- Với dữ liệu ảnh, kết quả thường là 10:1. Một số phương pháp còn cho kết quả cao hơn.
- Tạo ra sự hiệu quả trong việc lưu trữ và truyền tải dữ liệu.

#### Tỉ lê nén

- Là một trong các đặc trưng quan trọng nhất của mọi phương pháp nén.
- Gọi  $n_1$  là dữ liệu ban đầu,  $n_2$  là dữ liệu sau khi nén. Tỉ lệ nén  $C_r$  được tính như sau:

$$C_r = \frac{n_1}{n_2}$$

• Dữ liệu dư thừa:  $D_r = 1 - \frac{1}{C_r}$ 

## Dung lượng ảnh

Cho một ảnh I có kích thước MxN. Gọi L là số mức xám của ảnh I. Dung lượng của ảnh được tính theo công thức:  $M*N*L_{tb}$ . Trong đó:

$$L_{tb} = \sum_{k=0}^{L-1} I(r_k) p(r_k)$$

 $I(r_k)$  là độ dài bit biểu diễn cho mức xám  $r_k$ .  $p(r_k)$  là xác suất của mức xám  $r_k$  trong ảnh I.  $p(r_k) = \frac{n_k}{M*N}$ 

## Dung lượng ảnh

Ví dụ: Giả sử ảnh 3 bit I có thống kê về các mức xám như bảng sau. Tính dung lương của ảnh I.

$r_k$	$n(r_k)$	$p(r_k)$	mã
0	113	0.051	000
1	139	0.063	001
2	142	0.064	010
3	145	0.066	011
4	181	0.082	100
5	105	0.047	101
6	52	0.023	110
7	1323	0.601	111

• Giá trị trung bình mỗi pixel:

$$L_{tb} = \sum_{k=0}^{L-1} I(r_k) p(r_k) = 3 \text{ bit}$$

Dung lượng của ảnh I:

$$C = L_{tb} * \sum_{k=0}^{L-1} n(r_k)$$
  
 $C = 3 * 2200 = 6600 \text{ bit}$ 

# Phân loại các phương pháp nén ảnh

Có nhiều cách phân loại các phương pháp nén khác nhau:

- Phương pháp nén không mất mát thông tin.
  - Ảnh nén sau khi giải nén có thể thu được dữ liệu ảnh ban đầu.
  - Tỉ lệ nén thấp từ 2 đến 10 lần.
- Phương pháp nén có mất mát thông tin.
  - Ánh nén sau khi giải nén không thu được dữ liệu như ảnh gốc ban đầu.
  - Tỉ lệ nén cao từ 10 đến 30 lần.

# Phân loại các phương pháp nén ảnh

Dựa vào quá trình phát triển của các phương pháp nén. Có thể chia các phương pháp nén thành hai nhóm:

- Các phương pháp nén thế hệ thứ nhất: Gồm các phương pháp mà mức độ tính toán là đơn giản, thí dụ việc lấy mẫu, gán từ mã....
- Các phương pháp nén thế hệ thứ hai: Dựa vào độ bão hòa của tỷ lệ nén.

# Một số phương pháp nén ảnh

- Phương pháp nén ảnh Huffman.
- Phương pháp nén ảnh Run-Length Encoding.
- Phương pháp nén ảnh LZW

- Được đề xuất bởi Huffman vào năm 1952.
- Là phương pháp dựa vào mô hình thống kê.
- Ý tưởng của phương pháp: gán cho các mức xám có tần suất cao trong ảnh một từ mã ngắn, các mức xám có tần suất thấp từ mã dài.

**Ví dụ:** Cho ảnh I 3 bit, kích thước 44x50 có các mức xám được biểu diễn bởi các mã trong bảng sau. Tính dung lượng ảnh I ứng với các mã đó.

$r_k$	$n(r_k)$	$p(r_k)$	mã 1	mã 2
0	113	0.051	000	1011
1	139	0.063	001	1010
2	142	0.064	010	1001
3	145	0.066	011	1000
4	181	0.082	100	110
5	105	0.047	101	1110
6	52	0.023	110	1111
7	1323	0.601	111	0

• Đối với mã 1, dung lượng ảnh I:

$$C_1 = L_{tb} * M * N = 3 * 44 * 50 = 6600 bit$$

- Đối với mã 2:
  - Độ dài trung bình mỗi pixel:

$$L_{tb} = \sum_{k=0}^{L-1} I(r_k) p(r_k) = 2.1 bit$$

- Dung lượng ảnh I:

$$C_2 = L_{tb} * M * N = 2.1 * 44 * 50 = 4620 bit$$



Đầu vào: Ảnh I có kích thước MxN và có K mức xám.

Đầu ra: Ảnh I\* được nén theo phương pháp Huffman.

Phương pháp này gồm 2 giai đoạn:

#### Giai đoạn 1: Xây dựng cây nhị phân Huffman

- Duyệt toàn bộ ảnh để tính xác suất các mức xám có trong ảnh.
- Khởi tạo rừng có K cây, mỗi cây chỉ có một nút gốc, mỗi nút gốc tương ứng với một mức xám và có trọng số là tần số (hoặc xác suất) của mức xám đó trong ảnh.
- Thực hiện bước lặp cho đến khi rừng chỉ còn 1 cây:
  - Chọn hai 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 trọng số của hai gốc tạo thành nó.

#### Giai đoạn 2: Xây dựng mã cho các mức xám

- Bắt đầu từ 2 nút con ở gốc, gán giá trị 0 cho nút có xác suất cao và giá trị 1 cho nút có xác suất thấp.
- Đi dần xuống các nút lá cho đến nút cuối cùng của cây để gián giá trị cho tất cả các nút trên cây.

Cho ảnh 3 bit I có các thống kê mức xám như sau. Thực hiện nén ảnh I theo phương pháp Huffman.

$r_k$	$n(r_k)$	$p(r_k)$	mã
0	113	0.051	000
1	139	0.063	001
2	142	0.064	010
3	145	0.066	011
4	181	0.082	100
5	105	0.047	101
6	52	0.023	110
7	1323	0.601	111

$r_k$	$p(r_k)$
7	0.601
4	0.082
3	0.066
2	0.064
1	0.063
0	0.051
5	0.047
6	0.023

$r_k$	$p(r_k)$	node 1
7	0.601	0.601
4	0.082	0.082
3	0.066	0.070
2	0.064	0.066
1	0.063	0.064
0	0.051	0.063
5	0.047	0.051
6	0.023	

$r_k$	$p(r_k)$	node 1	node 2
7	0.601	0.601	0.601
4	0.082	0.082	- 0.114
3	0.066	0.070	0.082
2	0.064	0.066	0.070
1	0.063	0.064	0.066
0	0.051	0.063	0.064
5	0.047	0.051	
6	0.023		

$r_k$	$p(r_k)$	node 1	node 2	node 3
7	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130
3	0.066	0.070	0.082	0.114
2	0.064	0.066	0.070	0.082
1	0.063	0.064	0.066	0.070
0	0.051	0.063	0.064	
5	0.047	0.051		
6	0.023			

$r_k$	$p(r_k)$	node 1	node 2	node 3	node 4
7	0.601	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130	0.152
3	0.066	0.070	0.082	0.114	0.130
2	0.064	0.066	0.070	0.082	0.114
1	0.063	0.064	0.066	0.070	
0	0.051	0.063	0.064		
5	0.047	0.051			
6	0.023				

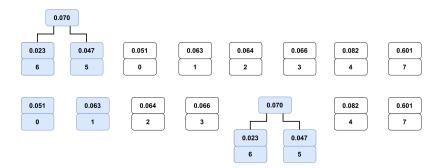
$r_k$	$p(r_k)$	node 1	node 2	node 3	node 4	node 5
7	0.601	0.601	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130	0.152	0.244
3	0.066	0.070	0.082	0.114	0.130	0.152
2	0.064	0.066	0.070	0.082	0.114	
1	0.063	0.064	0.066	0.070		
0	0.051	0.063	0.064			
5	0.047	0.051				
6	0.023					

$r_k$	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601	0.601	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130	0.152	0.244	0.396
3	0.066	0.070	0.082	0.114	0.130	0.152	
2	0.064	0.066	0.070	0.082	0.114		
1	0.063	0.064	0.066	0.070			
0	0.051	0.063	0.064				
5	0.047	0.051					
6	0.023						

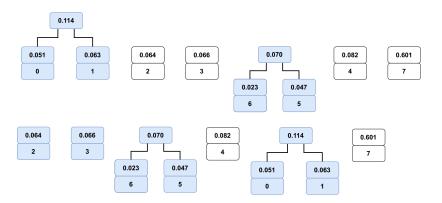
Khởi tạo rừng có 8 cây theo các mức xám ban đầu:



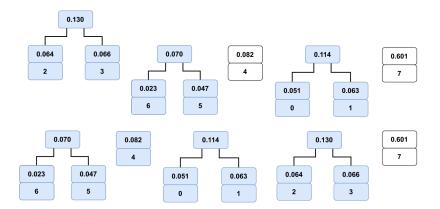
Vòng lặp 1: Thực hiện tạo cây mới từ 2 nút có trọng số nhỏ nhất:



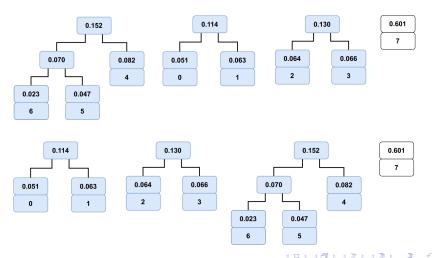
Vòng lặp 2: Thực hiện tạo cây mới từ 2 nút có trọng số nhỏ nhất:



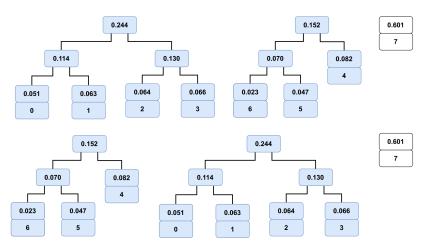
#### Vòng lặp 3: Thực hiện tạo cây mới từ 2 nút có trọng số nhỏ nhất:



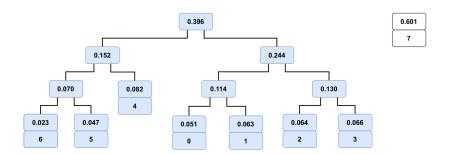
Vòng lặp 4: Thực hiện tạo cây mới từ 2 nút có trọng số nhỏ nhất:



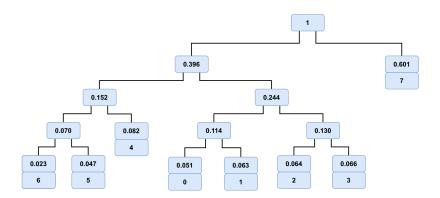
Vòng lặp 5: Thực hiện tạo cây mới từ 2 nút có trọng số nhỏ nhất:



Vòng lặp 6: Thực hiện tạo cây mới từ 2 nút có trọng số nhỏ nhất:



Vòng lặp 7: Thực hiện tạo cây mới từ 2 nút có trọng số nhỏ nhất:



$r_k$	$p(r_k)$	node 1	node 2		node 3	node 4	node 5	node 6	i
7	0.601	0.601	0.601		0.601	0.601	0.601	0.601	0
4	0.082	0.082	0.114		0.130	0.152	0.244	0.396	1
3	0.066	0.070	0.082		0.114	0.130	0.152		
2	0.064	0.066	0.070		0.082	0.114			
1	0.063	0.064	0.066	/	0.070				
0	0.051	0.063	0.064						
5	0.047	0.051							
6	0.023								

$r_k$	$p(r_k)$	node 1	no	ode 2		node 3		node 4	node 5		node 6	
7	0.601	0.601	0.	.601		0.601		0.601	0.601	0	0.601	0
4	0.082	0.082	_ 0.	.114		0.130		0.152	0.244	10	0.396	1
3	0.066	0.070	/ o.	.082		0.114		0.130	0.152	11		
2	0.064	0.066	0.	.070	/	0.082	$\cup$	0.114				
1	0.063	0.064	0.	.066	/	0.070						
0	0.051	0.063	0.	.064								
5	0.047	0.051										
6	0.023											

$r_k$	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601	0.601	0.601	0.601 0	0.601 0	0.601 0
4	0.082	0.082	0.114	0.130	0.152 11	0.244 10	0.396 <b>1</b>
3	0.066	0.070	0.082	0.114	0.130 100	0.152 11	
2	0.064	0.066	0.070	0.082	0.114 101		
1	0.063	0.064	0.066	0.070			
0	0.051	0.063	0.064				
5	0.047	0.051					
6	0.023						

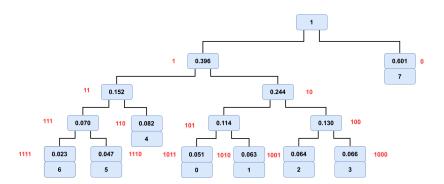
```
r_k
      p(r_k)
                      node 1
                                    node 2
                                                                 node 4
                                                                               node 5
                                                                                              node 6
                                                   node 3
7
       0.601
                      0.601
                                    0.601
                                                   0.601
                                                                 0.601
                                                                                0.601
                                                                                              0.601
4
       0.082
                      0.082
                                    0.114
                                                   0.130
                                                          100
                                                                 0.152
                                                                                0.244 10
                                                                                              0.396 1
3
       0.066
                      0.070
                                    0.082
                                                   0.114
                                                          101
                                                                 0.130 100
                                                                                0.152 11
2
                                    0.070
                                                   0.082 110
                                                                 0.114 101
       0.064
                      0.066
1
       0.063
                      0.064
                                    0.066
                                                   0.070 111
                                    0.064
0
       0.051
                      0.063
                      0.051
5
       0.047
6
       0.023
```

```
p(r_k)
r_k
                     node 1
                                   node 2
                                                                node 4
                                                  node 3
                                                                              node 5
                                                                                            node 6
7
       0.601
                     0.601
                                    0.601
                                                  0.601 0
                                                                0.601 0
                                                                              0.601 0
                                                                                            0.601 0
       0.082
                     0.082
                                    0.114
                                           101
                                                  0.130
                                                        100
                                                                0.152
                                                                              0.244 10
                                                                                            0.396 1
3
       0.066
                     0.070
                                    0.082
                                           110
                                                  0.114
                                                        101
                                                                0.130 100
                                                                              0.152 11
2
       0.064
                     0.066
                                    0.070
                                           11/1
                                                  0.082 110
                                                                0.114 101
1
       0.063
                     0.064
                                    0.066
                                           1000
                                                  0.070 111
0
       0.051
                     0.063
                                    0.064 1001
5
       0.047
                     0.051
       0.023
6
```

```
p(r_k)
r_k
                     node 1
                                   node 2
                                                 node 3
                                                               node 4
                                                                             node 5
                                                                                           node 6
7
       0.601
                     0.601
                                   0.601
                                                 0.601 0
                                                               0.601 0
                                                                             0.601 0
                                                                                           0.601 0
       0.082
                     0.082
                            110
                                   0.114
                                          101
                                                 0.130
                                                        100
                                                               0.152 11
                                                                             0.244 10
                                                                                           0.396 1
                                                               0.130 100
       0.066
                     0.070
                            111
                                   0.082
                                          110
                                                 0.114
                                                        101
                                                                             0.152 11
3
                     0.066
                            1000
                                   0.070
                                                 0.082 110
                                                               0.114 101
2
       0.064
                                          11/1
       0.063
                            1001
                                   0.066 1000
                                                 0.070 111
1
                     0.064
0
       0.051
                     0.063 1010
                                   0.064 1001
                     0.051 1011
5
       0.047
       0.023
6
```

```
p(r_k) code
r_k
                     node 1
                                   node 2
                                                 node 3
                                                               node 4
                                                                             node 5
                                                                                            node 6
7
       0.601
                     0.601
                                   0.601
                                          0
                                                 0.601 0
                                                                0.601
                                                                              0.601
                                                                                            0.601 0
       0.082
                     0.082
                                   0.114
                                                 0.130
                                                               0.152
                                                                              0.244 10
                                                                                            0.396 1
4
              110
                            110
                                          101
                                                        100
                                   0.082
                                                               0.130 100
3
       0.066
              1000
                     0.070
                            111/
                                          110
                                                 0.114
                                                        101
                                                                              0.152 11
                                   0.070
2
       0.064
              1001
                     0.066
                            1000
                                          11/1
                                                 0.082 110
                                                               0.114 101
       0.063
              1010
                     0.064
                            1001
                                   0.066 1000
                                                 0.070 111
1
0
       0.051
              1011
                     0.063 1010
                                   0.064 1001
5
       0.047 1110
                     0.051 1011
6
       0.023 1111
```

Gán mã cho cây Huffman như sau:



Dung lượng của ảnh I trước khi nén:

$$C_1 = L * M * N = 3 * 44 * 50 = 6600 bit$$

Giá trị trung bình biểu diễn cho mỗi mức xám là:

$$L_{tb} = \sum_{k=0}^{L-1} I(r_k) p(r_k) = 2.1 bit$$

Dung lượng của ảnh I sau khi nén:

$$C = L_{tb} * \sum_{k=0}^{L-1} n(r_k) = 2.1 * 2200 = 4620 bit$$

Đầu vào: Cho dãy bit thu được sau khi nén và cây Huffman.

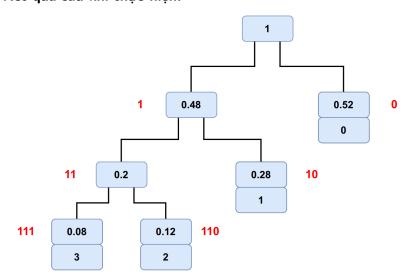
Đầu ra: Cho dãy mức xám thu được sau khi giải nén .

#### Cách thực hiện:

Duyệt dãy bit đã mã hóa và duyệt cây Huffman bắt đầu từ nút gốc với quy tắc: Nếu đọc được bit '0' sẽ chuyển sang cây con trái và nếu đọc được bit '1' sẽ chuyển sang cây con phải (giống như quy ước trong quá trình nén). Khi đạt đến nút lá thì ngắt phần chuỗi nhị phân đã đọc ra và thay bằng kí tự đại diện của nút lá. Chuỗi bit mã hóa sau khi bị ngắt lại tiếp tục lặp lại quá trình tương tự cho đến khi đọc hết chuỗi bit đầu vào.

**Bài tập áp dụng:** Cho ảnh I 2 bit, kích thước 5 x 5 được biểu diễn bởi ma trận sau. Hãy nén ảnh I theo phương pháp Huffman, tính dung lượng ảnh I trước và sau khi nén.

#### Kết quả sau khi thực hiện:



#### Kết quả sau khi thực hiện:

$r_k$	$n(r_k)$	$p(r_k)$	mã 1	mã 2
0	13	0.52	00	0
1	7	0.28	01	10
2	3	0.12	10	110
3	2	0.08	11	111

Dung lượng ảnh I trước khi nén:

$$C_1 = L_{tb} * M * N = 2 * 5 * 5 = 50bit$$

Dung lương ảnh I sau khi nén:

$$C_2 = L_{tb} * M * N = 1.68 * 5 * 5 = 42bit$$



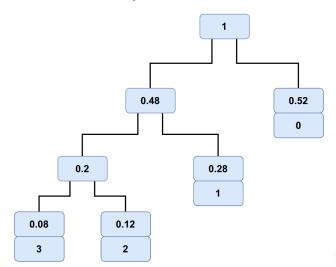
Đầu vào: Cho dãy bit thu được sau khi nén và cây Huffman.

Đầu ra: Cho dãy mức xám thu được sau khi giải nén .

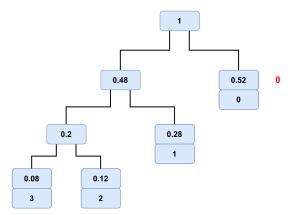
#### Cách thực hiện:

Duyệt dãy bit đã mã hóa và duyệt cây Huffman bắt đầu từ nút gốc với quy tắc: Nếu đọc được bit '0' sẽ chuyển sang cây con trái và nếu đọc được bit '1' sẽ chuyển sang cây con phải (giống như quy ước trong quá trình nén). Khi đạt đến nút lá thì ngắt phần chuỗi nhị phân đã đọc ra và thay bằng kí tự đại diện của nút lá. Chuỗi bit mã hóa sau khi bị ngắt lại tiếp tục lặp lại quá trình tương tự cho đến khi đọc hết chuỗi bit đầu vào.

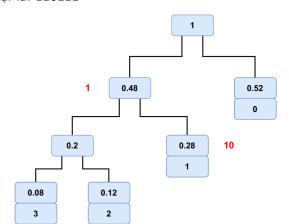
Đầu vào: 010110111 và cây Huffman



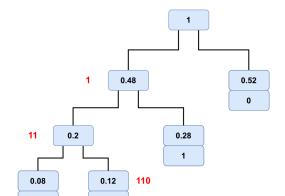
**Vòng lặp 1:** Từ nút gốc, đọc kí tự '0' từ dãy bit đầu vào và chuyển sang cây con bên phải. Vì đây là nút lá nên ngắt bit '0' gán cho mức xám số 0. Dãy bit hiện tại là: 10110111



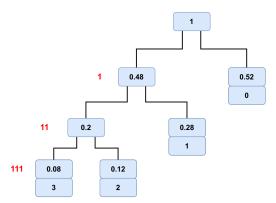
**Vòng lặp 2:** Từ nút gốc, đọc kí tự '1' từ dãy bit đầu vào và chuyển sang cây con bên trái, đọc tiếp kí tự '0' và chuyển sang cây con bên phải. Vì đây là nút lá nên ngắt bit '01' gán cho mức xám số 1. Dãy bit hiên tai là: 110111



**Vòng lặp 3:** Từ nút gốc, đọc kí tự '1' từ dãy bit đầu vào và chuyển sang cây con bên trái, đọc tiếp kí tự '1' và chuyển sang cây con bên trái, đọc tiếp kí tự '0' và chuyển sang cây con bên phải. Vì đây là nút lá nên ngắt bit '110' gán cho mức xám số 2. Dãy bit hiện tại là: 111

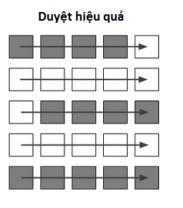


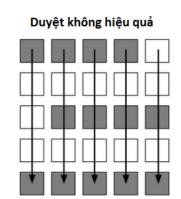
Vòng lặp 4: Từ nút gốc, đọc kí tự '1' từ dãy bit đầu vào và chuyển sang cây con bên trái, đọc tiếp kí tự '1' và chuyển sang cây con bên trái, đọc tiếp kí tự '1' và chuyển sang cây con bên trái. Vì đây là nút lá nên ngắt bit '111' gán cho mức xám số 3. Dãy bit hiện tại là: 'Null'



- Được đề xuất từ những năm 1950, lúc đầu được phát triển cho nén ảnh đen trắng.
- Nguyên tắc của phương pháp là phát hiện một loạt các bít lặp lại. Sau đó, thay thế chuỗi cũ bởi một chuỗi mới gồm 2 thông tin: chiều dài chuỗi và bít lặp (ký tự lặp).
- Phương pháp này chỉ có hiệu quả khi chiều dài dãy lặp lớn hơn một ngưỡng nào đó.
- Chuỗi thay thế sẽ có chiều dài ngắn hơn chuỗi cần thay.
- Phương pháp này được sử dụng trong việc mã hóa lưu trữ các ảnh Bitmap theo dạng PCX, BMP.

• Đối với ảnh, có thể có các thứ tư duyệt sau:





**Ví dụ:** Cho ảnh nhị phân I có ma trận ảnh  $(10 \times 10)$  như sau. Dùng phương pháp mã hóa mạch dài để nén ảnh. Cho biết tỷ số nén.

- Ánh trước khi nén có kích thước là 100 bit.
- Sử dụng mã hóa mạch dài, ta có chuỗi sau khi nén: (23,0), (4,1), (6,0), (1,1), (2,0), (1,1), (6,0), (4,1), (9,0), (1,1),(9.0), (1,1), (6,0), (4,1), (23,0).
- Có tất cả 15 cặp. Dùng 5 bit để mã hóa số lượng các bit lặp, dùng 1 bit để mã hóa 2 giá trị 0 hoặc 1. Như vậy, độ dài sau khi mã hóa là: 15\*(5+1) = 90 bit.
- Tỷ số nén:  $C_r = \frac{100}{90} = 1.1$
- Dữ liệu dư thừa:  $D_r = 1 \frac{1}{C_r} = 1 \frac{90}{100} = 0.1 = 10\%$

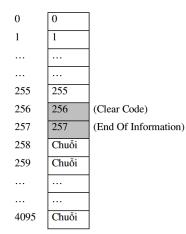
**Bài tập áp dụng:** Cho ánh xám 3 bit I kích thước 8 x 8 có ma trận như sau. Dùng phương pháp mã hóa mạch dài để nén ảnh. Cho cho biết tỷ số nén.

- Ånh trước khi nén có kích thước là 8\*8\*3 = 192 bit.
- Sử dung mã hóa mach dài, ta có chuỗi sau khi nén: (2,0), (9,1), (9,2), (4,3), (2,4), (3,5), (12,6), (8,7), (1,0), (1,7),(1,0), (1,7), (1,0), (1,7), (9,0).
- Có tất cả 15 cặp. Dùng 4 bit để mã hóa số lượng các bit lặp, dùng 3 bit để mã hóa các mức xám từ 0 đến 7. Như vậy, độ dài sau khi mã hóa là: 15\*(4+3) = 105 bit.
- Tỷ số nén:  $C_r = \frac{192}{105} = 1.829$
- Dữ liệu dư thừa:  $D_r = 1 \frac{1}{C_r} = 1 \frac{105}{192} = 0.453 = 45.3\%$

- Khái niệm nén dựa trên từ điển được Jocob Lempe và Abraham Ziv đưa ra vào năm 1977.
- Năm 1984 Welch đã cải tiến giải thuật LZ thành giải thuật hiệu quả hơn đặt tên là LZW. Là phương pháp nén không mất thông tin.
- Phương pháp này xây dựng từ điển các chuỗi pixel có tần suất lặp lại cao và thay thế bằng từ mã mỗi khi lặp lại chúng.
- LZW được dùng là chuẩn nén cho các dạng ảnh GIF và TIFF.

- Phần quan trọng nhất của phương pháp nén này là phải tạo một mảng rất lớn dùng để lưu giữ các chuỗi pixel đã gặp, mảng này được gọi là "Từ điển".
- Từ mã từ 0 đến 255 chứa các số nguyên từ 0 đến 255.
- Từ mã thứ 256 là mã đặc biệt. Để khắc phục trường hợp số mẫu lặp trong ảnh lớn hơn 4096. Mã xóa sẽ cho biết việc kết thúc mã hóa với từ điển cũ và bắt đầu bộ tự điển mới.
- Từ mã thứ 257 chứa mã kết thúc thông tin. Giúp phân chia file thành nhiều cum ảnh.
- Các từ mã còn lại từ 258 đến 4095 là các mẫu lặp lại trong ảnh.

#### Cấu trúc từ điển:



- 0 đến 255 được biểu diễn bởi 8 bit.
- 256 đến 511 được biểu diễn bởi 9 bit.
- 512 đến 1023 được biểu diễn bởi 10 bit.
- 1024 đến 2047 được biểu diễn bởi 11 bit.
- 2048 đến 4095 được biểu diễn bởi 12 bit.

#### Quá trình nén ảnh:

Bước 1: Đọc ảnh, lưu ma trận ảnh vào mảng 1 chiều

Bước 2: Khởi tạo dãy pixel S, từ điển D và dãy pixel đầu ra O

Duyêt mảng chứa các pixel ảnh

- Nếu pixel đang xét là dấu hiệu kết thúc
  - Đẩy các pixel của S ra ngoài (gán cho dãy O)
- Ngược lai đẩy pixel đó vào cuối dãy S
  - Nếu S chứa pixel chưa có trong từ điển D
    - Thêm S vào trong từ điển D
    - Đẩy pixel (hoặc dãy pixel trừ pixel cuối có trong dãy S) ra ngoài (gán cho dãy O)
  - Nếu S chứa pixel đã có trong từ điển
    - Quay lên, tiếp tục vòng lặp

**Ví dụ:** Cho ảnh xám I 8 bit kích thước 3 x 5 có ma trận như sau. Dùng phương pháp LZW để nén ảnh I. Cho biết tỷ số nén.

$$I = \begin{bmatrix} 72 & 69 & 76 \\ 76 & 79 & 72 \\ 69 & 76 & 76 \\ 79 & 72 & 69 \\ 76 & 76 & 79 \end{bmatrix}$$

- Ånh trước khi nén có dung lượng 3\*5\*8 = 120 bit.
- Chuỗi mức xám của ảnh ban đầu:
   72-69-76-76-79-72-69-76-76-79-72-69-76-79-79.
- Xây dựng từ điển theo LZW như sau:

Stt	Dãy hiện tại	Pixel kế tiếp	Từ	điển	Đầu ra
			Từ	Mã	
1	Null	72			
2	72	69	72-69	258	72
3	69	76	69-76	259	69
4	76	76	76-76	260	76
5	76	79	76-79	261	76
6	79	72	79-72	262	79
7	72	69	72-69	đã có 258	
8	72-69	76	72-69-76	263	72-69 (258)
9	76	76	76-76	đã có 260	
10	76-76	79	76-76-79	264	76-76 (260)
11	79	72	79-72	đã có 262	
12	79-72	69	79-72-69	265	79-72 (262)
13	69	76	69-76	đã có 259	
14	69-76	76	69-76-76	266	69-76 (259)
15	76	79	76-79	đã có 261	
16	76-79	#			76-79 (261)

- Chuỗi mức xám của ảnh ban đầu: 72-69-76-76-79-72-69-76-76-79-72-69-76-79.
- Chuỗi mức xám của ảnh ban đầu sau khi nén như sau: 72-69-76-76-79-258-260-262-259-261.
- Dung lượng sau khi nén: 5\*8 + 5\*9 = 85 bit.
- Tỷ số nén:  $C_r = \frac{120}{85} = 1.412$
- Dữ liệu dư thừa:  $D_r = 1 \frac{1}{C_r} = 1 \frac{85}{120} = 0.2916 = 29.17\%$

#### Quá trình giải nén:

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
	Day men en	r ixer ite itep	Từ	Mã	244.4
1	Null	72			
2	72	69	72-69	258	72
3	69	76	69-76	259	69
4	76	76	76-76	260	76
5	76	79	76-79	261	76
6	79	258	79-72	262	79
7	258	260	72-69-76	263	72-69
8	260	262	76-76-79	264	76-76
9	262	259	79-72-69	265	79-72
10	259	261	69-76-76	266	69-76
11	261	#			76-79

Bài tập áp dụng 1: Cho ánh xám 8 bit I kích thước 5 x 5 có ma trận như sau. Dùng phương pháp LZW để nén ảnh. Cho cho biết tỷ số nén, dữ liêu dư thừa.

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	72			
2	72	79	72-79	258	72
3	79	74	79-74	259	79
4	74	76	74-76	260	74
5	76	72	76-72	261	76
6	72	79	72-79	đã có 258	
7	72-79	74	72-79-74	262	72-79 (258)
8	74	76	74-76	đã có 260	
9	74-76	72	74-76-72	263	74-76 (260)
10	72	79	72-79	đã có 258	
11	72-79	74	72-79-74	đã có 262	
12	72-79-74	76	72-79-74-76	264	72-79-74 (262)
13	76	72	76-72	đã có 261	
14	76-72	79	76-72-79	265	76-72 (261)
15	79	74	79-74	đã có 259	
16	79-74	76	79-74-76	266	79-74 (259)
17	76	72	76-72	đã có 261	
18	76-72	79	76-72-79	đã có 265	
19	76-72-79	74	76-72-79-74	267	76-72-79 (265)
20	74	76	74-76	đã có 260	
21	74-76	72	74-76-72	đã có 263	
22	74-76-72	79	74-76-72-79	268	74-76-72 (263)
23	79	74	79-74	đã có 259	
24	79-74	76	79-74-76	đã có 266	
25	79-74-76	72	79-74-76-72	269	79-74-76 (266)
26	72	#			72

- Chuỗi mức xám của ảnh ban đầu:
   72-79-74-76-72-79-74-76-72-79-74-76-72-79-74-76-72-79-74-76-72.
- Chuỗi mức xám của ảnh ban đầu sau khi nén như sau:
   72-79-74-76-258-260-262-261-259-265-263-266-72.
- Dung lượng ảnh trước khi nén: 5\*5\*8 = 200 bit.
- Dung lượng ảnh sau khi nén: 5\*8 + 8\*9 = 112 bit.
- Tỷ số nén:  $C_r = \frac{200}{112} = 1.786$
- Dữ liệu dư thừa:  $D_r = 1 \frac{1}{C_r} = 1 \frac{112}{200} = 0.44 = 44\%$

#### Quá trình giải nén:

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	72			
2	72	79	72-79	258	72
3	79	74	79-74	259	79
4	74	76	74-76	260	74
5	76	258	76-72	261	76
6	258	260	72-79-74	262	72-79
7	260	262	74-76-72	263	74-76
8	262	261	72-79-74-76	264	72-79-74
9	261	259	76-72-79	265	76-72
10	259	265	79-74-76	266	79-74
11	265	263	76-72-79-74	267	76-72-79
12	263	266	74-76-72-79	268	74-76-72
13	266	#			79-74-76

Bài tập áp dụng 2: Cho ánh xám 8 bit I kích thước 4 x 4 có ma trân như sau. Dùng phương pháp LZW để nén ảnh. Cho cho biết tỷ số nén, dữ liêu dư thừa.

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	39			
2	39	39	39-39	258	39
3	39	126	39-126	259	39
4	126	126	126-126	260	126
5	126	39	126-39	261	126
6	39	39	39-39	đã có 258	
7	39-39	126	39-39-126	262	39-39 (258)
8	126	126	126-126	đã có 260	
9	126-126	39	126-126-39	263	126-126 (260)
10	39	39	39-39	đã có 258	
11	39-39	126	39-39-126	đã có 262	
12	39-39-126	126	39-39-126-126	264	39-39-126 (262)
13	126	39	126-39	đã có 261	
14	126-39	39	126-39-39	265	126-39 (261)
15	39	126	39-126	đã có 259	
16	39-126	126	39-126-126	266	39-126 (259)
17	126	#			126

- Chuỗi mức xám của ảnh ban đầu: 39-39-126-126-39-39-126-126-39-39-126-126-39-39-126-126.
- Chuỗi mức xám của ảnh ban đầu sau khi nén như sau: 39-39-126-126-258-260-262-261-259-126.
- Dung lượng ảnh trước khi nén: 4\*4\*8 = 128 bit.
- Dung lượng ảnh sau khi nén: 5\*8 + 5\*9 = 85 bit.
- Tỷ số nén:  $C_r = \frac{128}{85} = 1.506$
- Dữ liệu dư thừa:  $D_r = 1 \frac{1}{C_r} = 1 \frac{85}{128} = 0.336 = 33.6\%$

#### Quá trình giải nén:

Stt	Dãy hiện tại	Pixel kế tiếp	Từ điển		Đầu ra
			Từ	Mã	
1	Null	39			
2	39	39	39-39	258	39
3	39	126	39-126	259	39
4	126	126	126-126	260	126
5	126	258	126-39	261	126
6	258	260	39-39-126	262	39-39
7	260	262	126-126-39	263	126-126
8	262	261	39-39-126-126	264	39-39-126
9	261	259	126-39-39	265	126-39
10	259	126	39-126-126	266	39-126
11	126	#			126