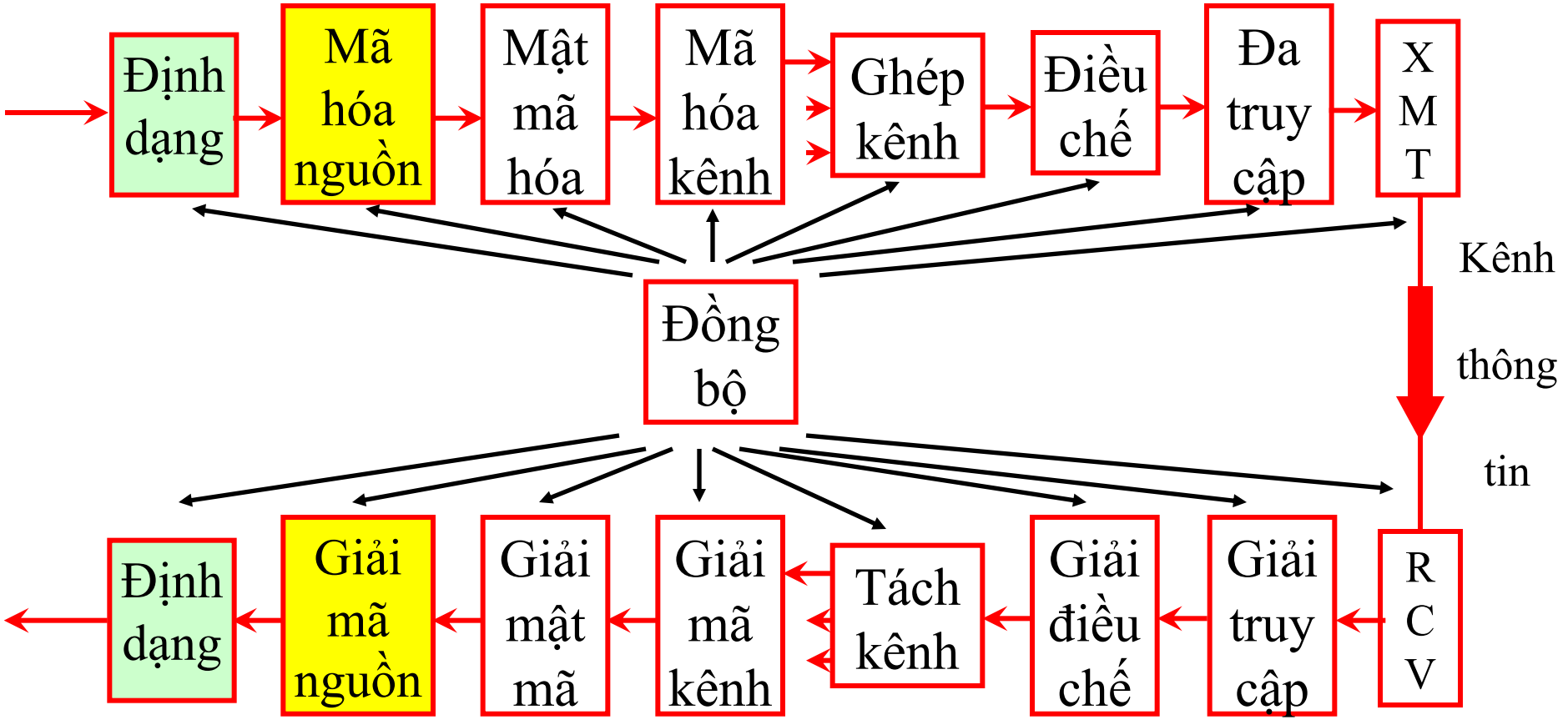
BÀI 4:

KỸ THUẬT MÃ HÓA NGUỒN

Hệ thống thông tin số điển hình



Nội dung bài 4

1. Giới thiệu kỹ thuật mã hóa nguồn
2. Kỹ thuật mã hóa Huffman
3. Một số ứng dụng của mã Huffman

Nội dung bài 4

1. **Giới thiệu kỹ thuật mã hóa nguồn**
2. Kỹ thuật mã hóa Huffman
3. Một số ứng dụng của mã Huffman

Cơ sở lý thuyết

- Hệ thống thông tin cơ bản:

- Các đại lượng đo:

- lượng tin (chứa trong một ký tự)
- entropy (lượng tin trung bình của nguồn tin)
- độ dư của nguồn tin
- tốc độ lập tin của nguồn tin: chỉ ra sự hình thành tin để đưa vào kênh
- dung lượng kênh: lượng tin tối đa kênh cho phép truyền qua trong một đơn vị thời gian

Cơ sở lý thuyết (tt)

- Công thức tính lượng tin chứa trong một ký tự nguồn:

$$I(i) = \log_2 \frac{1}{p(i)} = -\log_2 p(i)$$

- Lượng tin tỷ lệ nghịch với xác suất
- Nếu ký tự chắc chắn được sinh ra và truyền đi p(i) = 1 → việc truyền ký tự không có ý nghĩa gì cả → lượng tin bằng 0
- Đơn vị đo: bit (nếu cơ số 2)

Cơ sở lý thuyết (tt)

- Công thức tính entropy:

$$H = \sum_{i=1}^{M} p(i) \log_2 \frac{1}{p(i)} = -\sum_{i=1}^{M} p(i) \log_2 p(i)$$ [bit/ký tự]

- Entropy cực đại: ký hiệu Hmax – đạt được chỉ khi tất cả các ký tự nguồn đều được sinh ra với cùng xác suất
- Công thức tính độ dư của nguồn tin:

$$r = \frac{H_{\max} - H}{H_{\max}} = 1 - \frac{H}{H_{\max}}$$

Cơ sở lý thuyết (tt)

- Công thức tính tốc độ lập tin của nguồn:

$$R = n_0 H$$

n_0: số ký tự nguồn sinh ra trong một đơn vị thời gian **[ký tự/s]**

H: entropy **[bit/ký tự]**

→ Muốn tăng R, tăng entropy và cải thiện cấu trúc vật lý của nguồn tin

Cơ sở lý thuyết (tt)

- Công thức tính dung lượng kênh:

$$C = R_{\max} = n_0 H_{\max}$$

- Đối với kênh không nhiễu: dung lượng kênh bằng với lượng tin tối đa nguồn có thể lập được trong một đơn vị thời gian

Truyền tin trong kênh không nhiễu

- **Định lý Shannon:** Nếu R < C thì có thể mã hóa để làm cho tốc độ lập tin của nguồn tiếp cận với dung lượng kênh:

$$C - R < \varepsilon$$

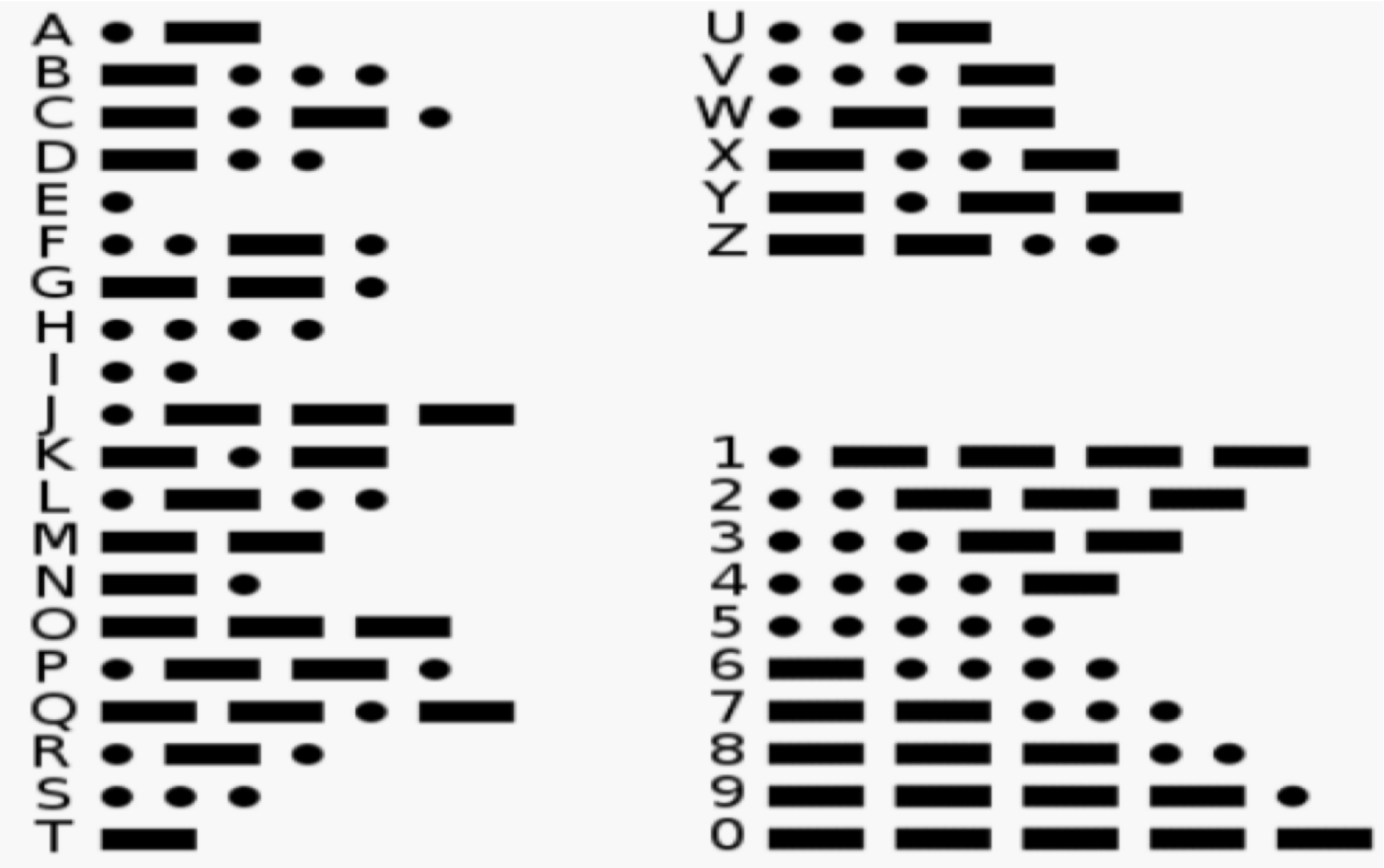
Phương pháp mã hóa này gọi là mã hoá nguồn

- Nguyên lý mã hóa nguồn: làm cho cấu trúc thống kê của nguồn trở nên hợp lý hơn bằng cách tăng entropy của các ký tự dùng để mã hóa nguồn
- Mã hóa nguồn loại bỏ độ dư của nguồn tin, tăng hiệu suất mã hóa

Mã hóa nguồn thống kê

- **Mã hóa thống kê:** mã hóa các ký tự có xác suất sinh ra lớn bằng các từ mã ngắn và ngược lại

Morse code



Mã hóa nguồn thống kê tối ưu

Độ dài từ mã trung bình nhỏ nhất (mã nén) → mã không đều, giữa các từ mã không có ký hiệu phân cách

$$\eta_s = \frac{H}{H_{\max}} \times 100\% = \frac{\sum_{m=1}^{M} p(m) \log_2 \frac{1}{p(m)}}{\log_2 M} \times 100\%$$

$$\eta_c = \frac{H}{L} \times 100\% = \frac{H}{\sum_{m=1}^{M} p(m) l(m)} \times 100\%$$

Ví dụ tính hiệu suất mã hóa

Ký tự	A	B	C	D	E	F	G	H
X.suất	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04

Entropy (rate) = **2.55 bit/ký tự**

Entropy (rate) cực đại = **3 bit/ký tự**

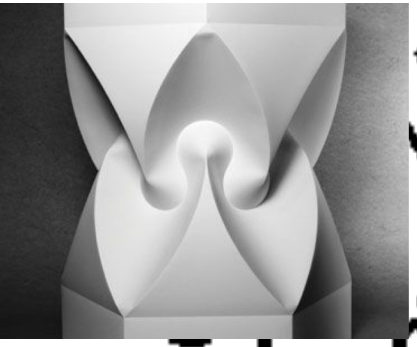
Nếu mã hóa nguồn tin trên dùng:

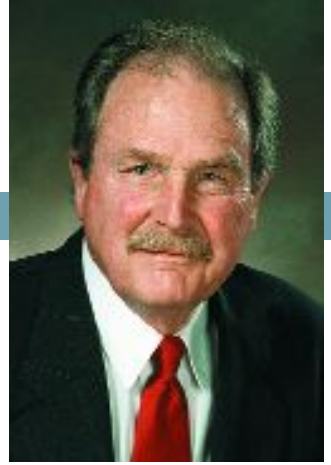
- 3 bit/ký tự → hiệu suất mã hóa 85%
- 2.55 bit/ký tự → hiệu suất mã hóa 100%
- < 2.55 bit/ký tự → có tổn hao

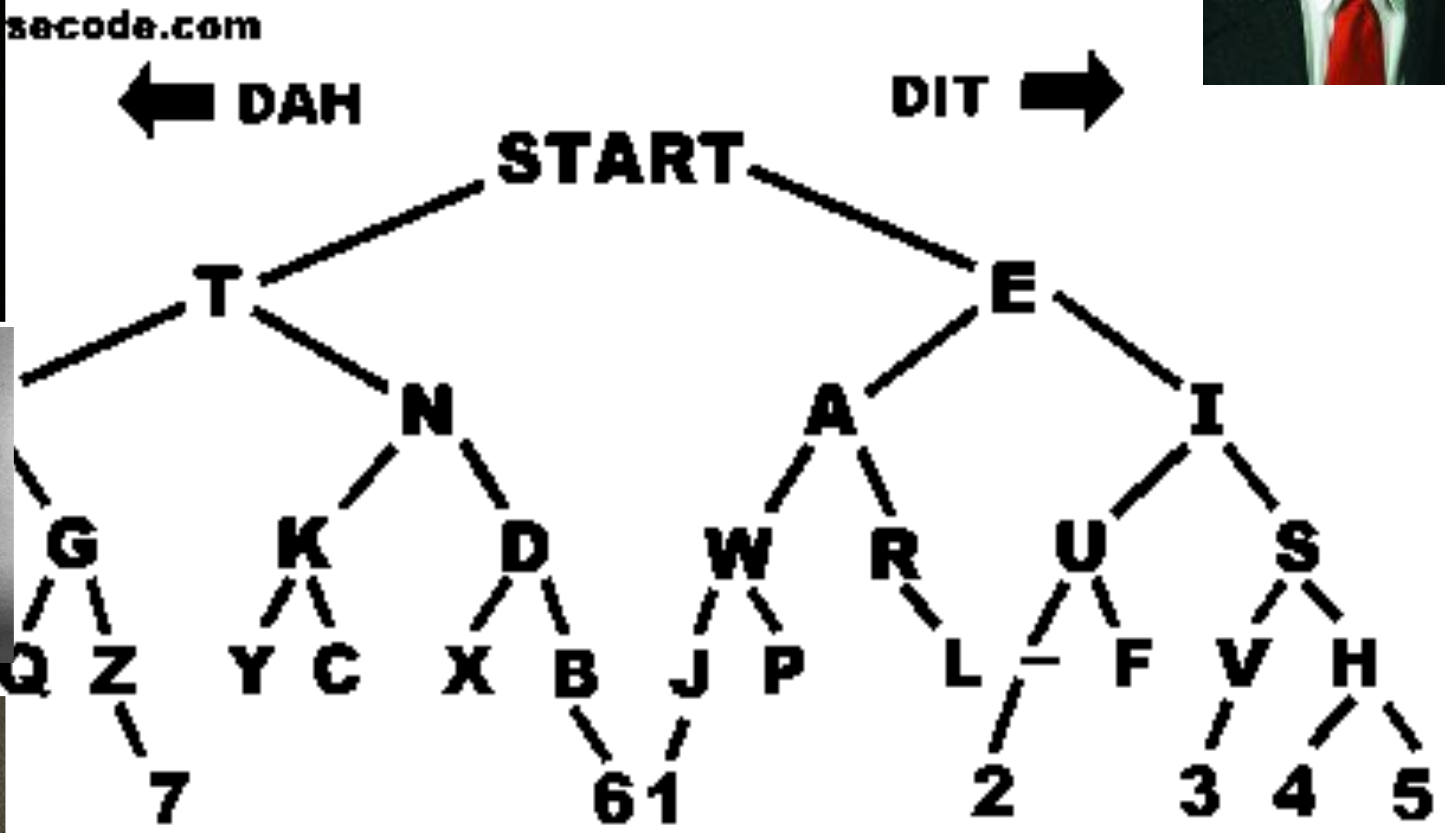
Nội dung bài 4

1. Giới thiệu kỹ thuật mã hóa nguồn
2. **Kỹ thuật mã hóa Huffman**
3. Một số ứng dụng của mã Huffman

HUFFMAN

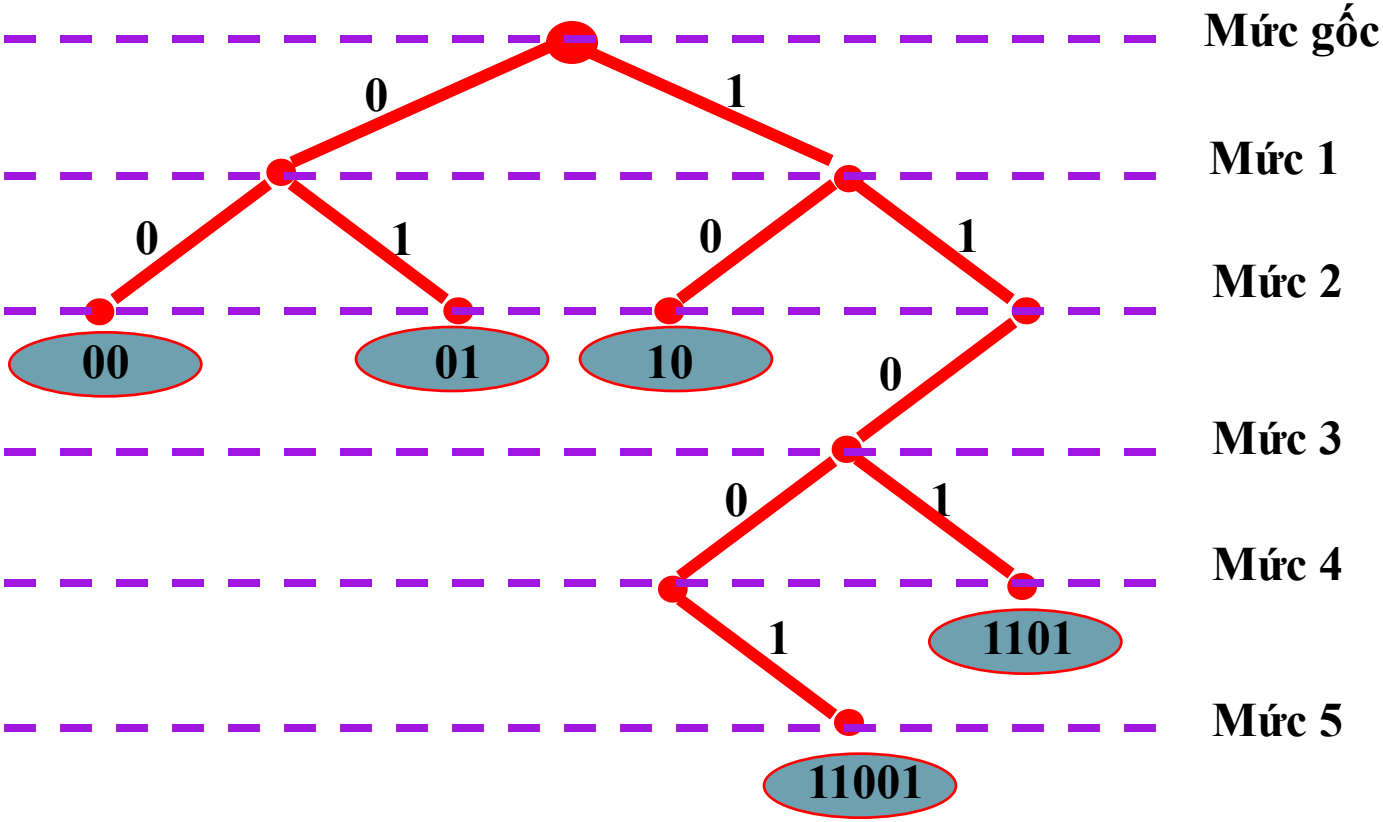





secode.com
DAH
DIT
START
T
E
N
A
I
G
K
D
W
R
U
S
Q
Z
Y
C
X
B
J
P
L
F
V
H
7
61
2
3
4
5

Biểu diễn mã bằng cây mã

Ví dụ: Bộ mã gồm các từ mã 00, 01, 10, 11001, 1101



Yêu cầu đối với mã: có tính chất **prefix** (từ mã ngắn hơn không trùng với phần đầu của từ mã dài hơn)

Giới thiệu mã Huffman

- **Tác giả:** David A. Huffman, một sinh viên sau đại học của MIT (1952)
- Là mã thống kê tối ưu
- Ký tự xuất hiện với xác suất lớn ←→ từ mã ngắn
- Ký tự xuất hiện với xác suất bé ←→ từ mã dài
- Mã hóa và giải mã dựa vào cây mã
- Không có khả năng phát hiện lỗi và sửa lỗi
- **Ứng dụng:** máy tính IBM, HDTV, modem, máy fax, định dạng file ZIP, JPEG...

Phân loại mã Huffman

- Dựa vào cây mã và từ mã cố định hay thay đổi
- **Có 2 loại mã Huffman:**

1. Mã Huffman cơ sở hay Huffman tĩnh (basic / static Huffman code)
2. Mã Huffman động (dynamic Huffman code)

Mã Huffman tĩnh

- Cây mã đã định trước
- Lập cây mã dựa trên cơ sở phân tích dữ liệu chuẩn
- Không cần lưu cây mã/ xác suất xuất hiện trong dữ liệu mã hóa
- Hiệu quả nén phụ thuộc vào sự phân bố xác suất các ký tự trong dữ liệu
- Thời gian mã hóa nhanh

Mã Huffman động

- Cây mã không cố định
- Lập cây mã dựa trên cơ sở phân tích động dữ liệu
- Phải lưu cây mã/ xác suất xuất hiện trong dữ liệu mã hóa (phần header)
- Hiệu quả nén cao hơn mã Huffman tĩnh
- Thời gian mã hóa chậm → chia dữ liệu ra từng đoạn, cập nhật cây mã theo chu kỳ

Thuật toán mã hóa Huffman

- Để mã hóa, cần phải phân tích dữ liệu vào để có thông tin về xác suất xuất hiện của các ký tự nguồn
- **Thuật toán mã hóa:**

1. Sắp xếp các ký tự nguồn theo thứ tự xác suất giảm dần.
2. Gán cho hai ký tự có xác suất thấp nhất với hai nhánh. Giảm từ 2 ký tự xuống còn 1 ký tự với xác suất bằng tổng của hai xác suất.
3. Lặp lại từ bước (1) cho đến khi chỉ còn lại một ký tự duy nhất với xác suất là **1**.
4. Duyệt cây mã để tìm ra bảng mã

Mã hóa Huffman tĩnh

- Trọng số của nút: xác suất của nút.
- Quy ước:
  - Nhánh đi ra từ ký hiệu có xác suất cao là nhánh 1;
  - Nhánh đi ra từ ký hiệu có xác suất thấp là nhánh 0;
  - Nhánh 0 vẽ bên trái;
  - Nhánh 1 vẽ bên phải.
- Cây mã Huffman tối ưu: Thứ tự trọng số tăng dần theo chiều từ dưới lên trên và từ trái sang phải.

Ví dụ mã hóa Huffman tĩnh

Bản tin chứa 8 ký tự từ A đến H, với xác suất xuất hiện:

Ký tự	A	B	C	D	E	F	G	H
X.suất	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04

Entropy của nguồn

$$H = \sum_{m=1}^{8} p(m) \log_2 \frac{1}{p(m)} = 2.55$$

Entropy cực đại

$$H_{max} = \log_2 8 = 3$$

Hiệu suất nguồn

$$\eta = \frac{2.55}{3} x100\,\% = 85\,\%$$

Ví dụ mã hóa Huffman tĩnh

Entropy của nguồn

$$H = \sum_{m=1}^{8} p(m) \log_2 \frac{1}{p(m)} = 2.55$$

Entropy cực đại

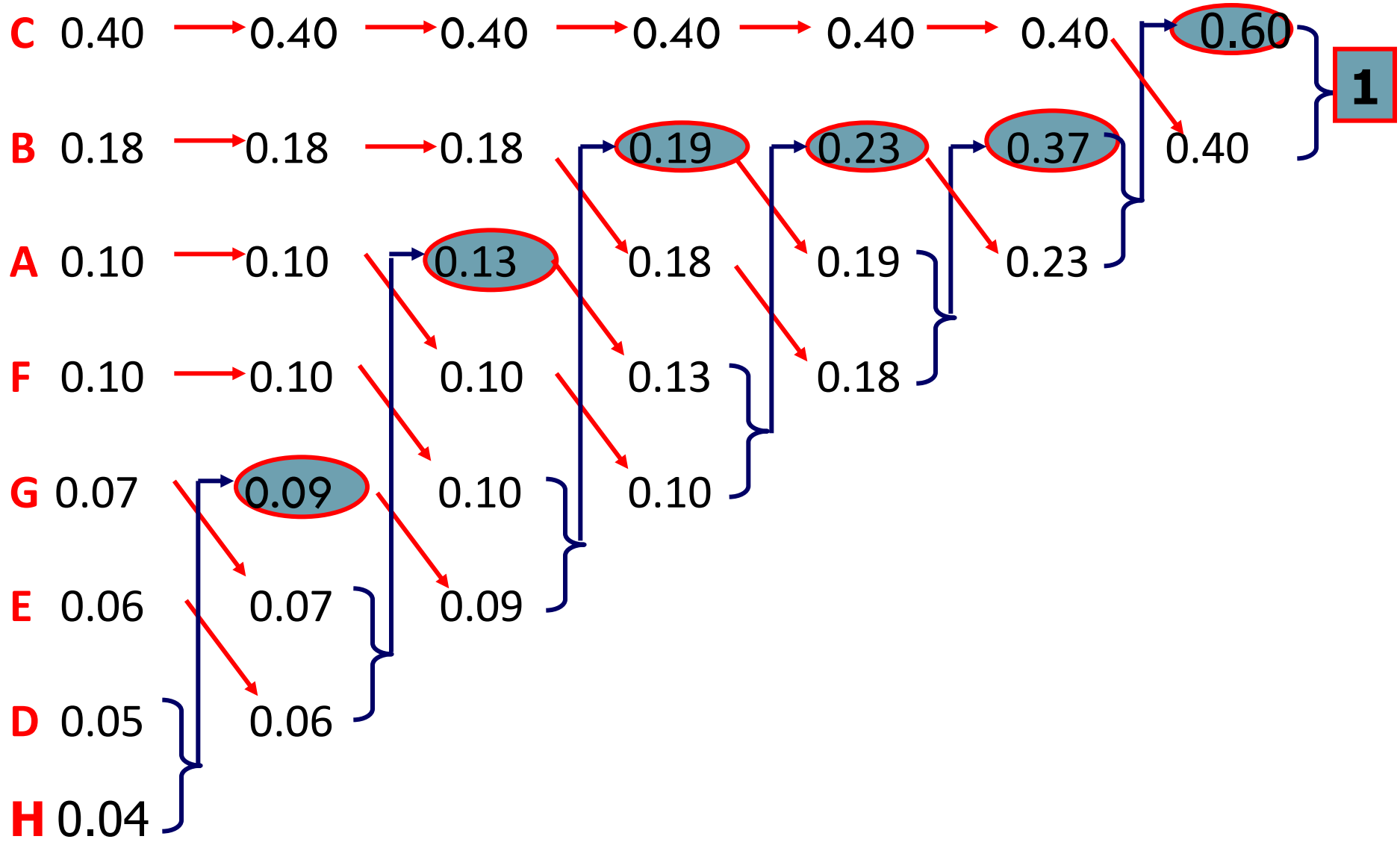
$$H_{max} = \log_2 8 = 3$$

Hiệu suất nguồn

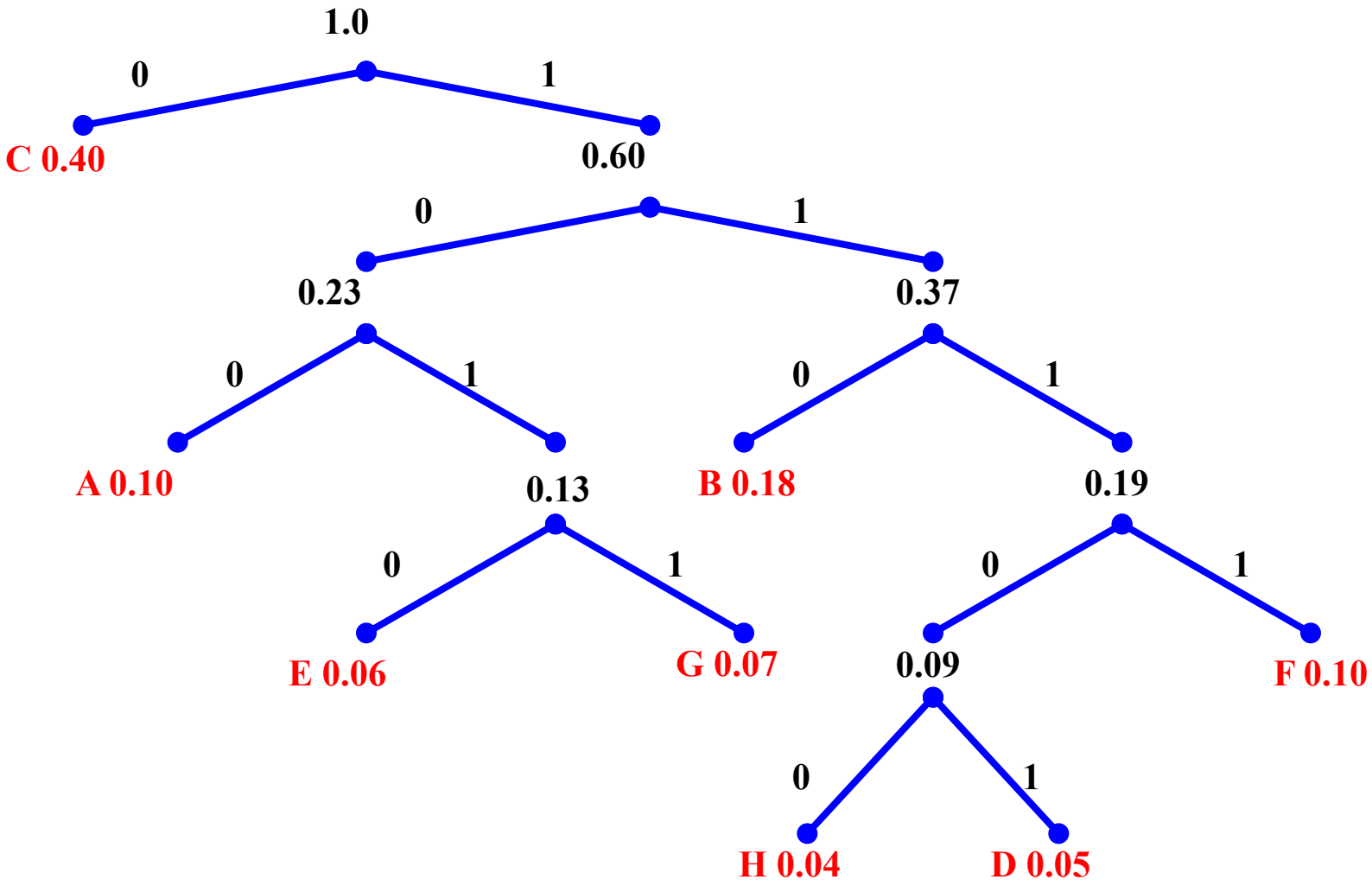
$$\eta = \frac{2.55}{3} x100\,\% = 85\,\%$$

- **Nếu mã hóa nguồn tin trên dùng 3 bit/ký tự thì hiệu suất mã hóa là 85%**
- **Nếu mã hóa nguồn tin trên dùng 2.55 bit/ký tự thì hiệu suất mã hóa là 100%**

Ví dụ mã hóa Huffman tĩnh (tt)



Ví dụ mã hóa Huffman tĩnh (tt)



Thứ tự trọng số = 0.04 0.05 0.06 0.07 0.09 0.10 0.10 0.13 0.18 0.19 0.23 0.37 0.40 0.60

Ví dụ mã hóa Huffman tĩnh (tt)

Bảng mã Huffman của nguồn tin trên:

Ký tự	A	B	C	D	E	F	G	H
X.suất	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04
Từ mã	100	110	0	11101	1010	1111	1011	11100

Độ dài từ mã trung bình:

$$L = 1(0.4) + 3(0.18 + 0.10) + 4(0.10 + 0.07 + 0.06) + 5(0.05 + 0.04)$$
$$= 2.61(\text{bit/ký tự})$$

Hiệu suất mã hóa:

$$\eta = \frac{H}{L} x100\,\% = \frac{2.55}{2.61} x100\,\% = 97.7\,\%$$

Ví dụ mã hóa Huffman tĩnh (tt)

Độ dài từ mã trung bình:

$$L = 1(0.4) + 3(0.18 + 0.10) + 4(0.10 + 0.07 + 0.06) + 5(0.05 + 0.04)$$
$$= 2.61(\text{bit/ký tự})$$

Hiệu suất mã hóa:

$$\eta = \frac{H}{L} x100\,\% = \frac{2.55}{2.61} x100\,\% = 97.7\,\%$$

Nếu mã hóa nguồn tin trên dùng mã Huffman

thì cần 2.61 bit/ký tự → hiệu suất mã hóa 97.7%

Hoạt động 5

Cho một nguồn tin với các ký tự và phân bố xác xuất như sau:

Ký tự	A	B	C	D	E	F	G	H
Xác suất	0.13	0.12	0.05	0.1	0.3	0.02	0.08	0.2

- Hãy tính entropy, entropy cực đại và hiệu suất của nguồn?
- Xây dựng cây mã Huffman tối ưu và xác định các từ mã tương ứng cho các ký tự. Tính hiệu suất của mã

Nội dung bài 4

1. Giới thiệu kỹ thuật mã hóa nguồn
2. Kỹ thuật mã hóa Huffman
3. **Một số ứng dụng của mã Huffman**

Ứng dụng mã hóa fax

- Fax là dịch vụ truyền các tài liệu số đen trắng
- Độ phân giải: *3.85 dòng/mm* và *1728 pixel/dòng*
- Mỗi pixel được lượng tử hóa nhị phân cho 2 màu
- Một trang giấy A4 tạo ra khoảng 2 triệu bit
- Fax nhóm 3 (ITU): tỷ lệ nén 10:1

Ứng dụng mã hóa fax (tt)

- Phân tích tổng quát trang tài liệu quét
- Các pixel trắng/đen liên tục tạo thành các run-length trắng/đen
- Độ dài run-length: số lượng pixel chứa trong run-length
- Run-length hay xuất hiện ←→ từ mã ngắn
- Run-length ít xuất hiện ←→ từ mã dài
- Bảng mã fax:
  - Bảng mã cuối: run-length 0 - 63 pixel, bước nhảy 1 pixel
  - Bảng mã make-up: run-length 64 - 2560 pixel, bước nhảy 64 pixel

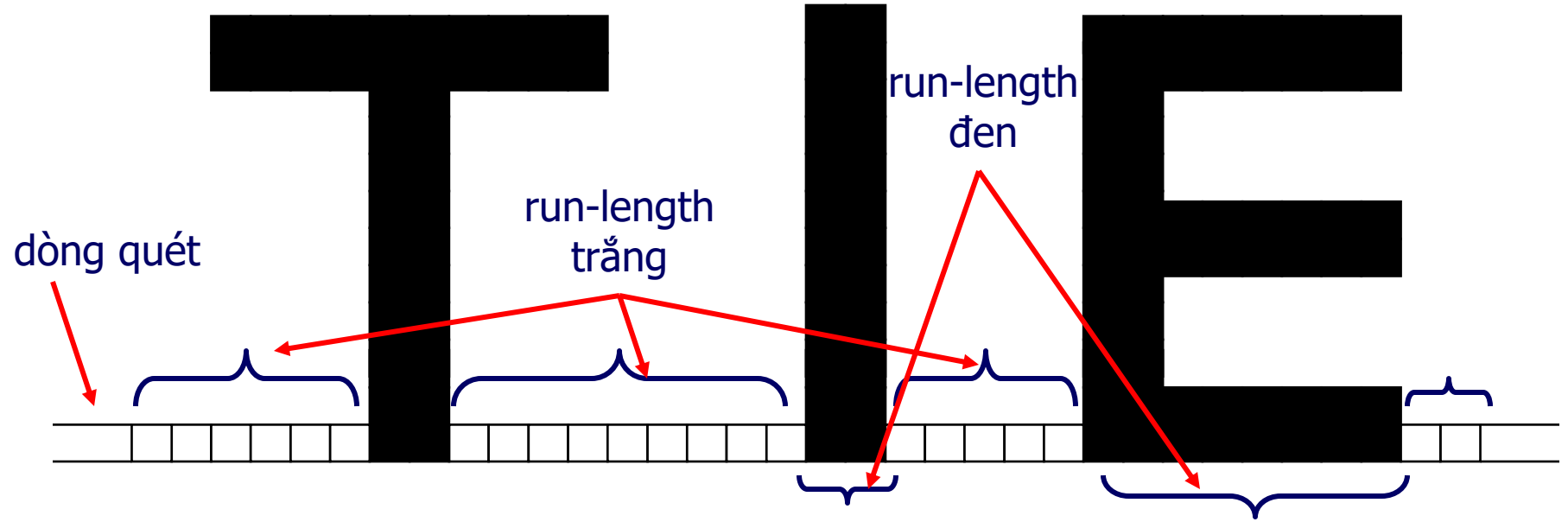
Bảng mã fax

Run length trắng	Từ mã	Run length đen	Từ mã
0	00110101	0	0000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
.....	.....	.....	.....
62	00110011	62	000001100110
63	00110100	63	000001100111

Run length trắng	Từ mã	Run length đen	Từ mã
64	11011	64	0000001111
128	10010	128	000011001000
192	010111	192	000011001001
.....	.....	.....	.....
2560	000000011111	2560	000000011111
EOL	00000000001	EOL	00000000001

Ví dụ: run-length trắng 131 pixel ←→ 10010 + 1000

Ví dụ mã hóa fax



34bit 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1

| 6W | 2B | 9W | 2B | 5W | 8B | 2W |

27 bit

1110	10100	1100	0111
11	11	000101	

Ứng dụng mã hóa JPEG

- JPEG = **J**oint **P**hotographic **E**xperts **G**roup (1992)
- Là phương pháp nén ảnh hiệu quả, hệ số nén vài chục lần
- Là phương pháp nén được dùng phổ biến hiện nay (máy ảnh số, www, điện thoại di động…)
- **Các bước mã hóa chính:**
  - Chia nhỏ bức ảnh thành khối 8x8
  - Biến đổi DCT (Discrete Cosine Transform)
  - Lượng tử hóa
  - Mã hóa Huffman)

JPEG

487x414 pixels,
Uncompressed, 600471 Bytes,24 bpp

487x414 pixels
41174 Bytes, 1.63 bpp, CR=14.7

Bài tập ôn giữa kỳ

1. Cho một nguồn tin với các ký tự và phân bố xác xuất như sau:

Ký tự	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
Xác suất	0.15	0.1	0.05	0.2	0.3	0.02	0.06	0.12

a. Hãy tính entropy, entropy cực đại và hiệu suất của nguồn?

b. Xây dựng cây mã Huffman tối ưu và xác định các từ mã tương ứng cho các ký tự. Tính hiệu suất của mã

2. Dải biên độ từ -4 đến 4 được lượng tử hóa đều sử dụng lần lượt 3 bit và 4 bit (-4 tương ứng với 000 (0000) và 4 tương ứng với 111 (1111)). Đối với từng trường hợp:

a. Hãy tính kích thước bước lượng tử.

b. Hãy tính sai số lượng tử hóa của các giá trị sau: **i. 3.55 ii. 1.235 iii. -0.34 iv. -2.12**

c. Nhận xét ảnh hưởng của kích thước bước lượng tử đến sai số lượng tử hóa

3. Cho bản tin nhị phân **1010 0000 0000 1101 0000 0110 0001**.

Hãy vẽ dạng sóng khi mã hóa bản tin trên sử dụng các loại mã hóa đường: Unipolar Non-Return-to-Zero (NRZ), Unipolar Return-to-Zero (RZ), Polar RZ, Bipolar RZ, HBD3.