٤٢٠٢ Digital Multimedia

Lecture 4

Dr. Shaimaa Othman

Entropy Encoding

- Basic Idea
  - Repeated code takes code of length inverse proportional to its repeat values
  - Two problems (code length?, code structure?)
- Mathematical
  - Studies for the amount of information carried out by symbols
  - Histograms are the base of the entropy as it has the repeations
  - Source codes are considered as symbols that represents events with probabilities that is mapped to information.

Entropy

- The number of bits needed to encode a media source is lower-bounded by its “Entropy”.

- ***Self information*** of an event A is defined as

$$-\log_b P(A)$$

where P(A) is the probability of event A.

If b equals 2, the unit is “bits” which is our case.

Example

- A source outputs two symbols (the alphabet has 2 symbols) 0 or 1. P(0) = 0.25, P(1) = 0.75.

  Information we get when receiving a 0 is

  log_2 (1/0.25) = 2 bit ;

  when receiving a 1 is

  log_2 (1/0.75) = 0.4150 bit .

Properties of Self Information

- The letter with smaller probability has high self information.

- The information we get when receiving two independent letters are summation of each of the self information.

$$
\begin{aligned}
&-\log_2 P(s_a, s_b) \\
&= -\log_2 P(s_a) P(s_b) \\
&= [-\log_2 P(s_a)] + [-\log_2 P(s_b)]
\end{aligned}
$$

Entropy

- An source has symbols {s1, s2, …, sn}, and the symbols are independent, the average self-information is

$$H= \sum_{1}^{n} P(s_i)\log_2(1/P(s_i)) \text{ bits}$$

- H is called the ***Entropy*** of the source.

- The number of bits per symbol needed to encode a media source is lower-bounded by its “Entropy”.

Entropy (cont)

- Example:

  A source outputs two symbols (the *alphabet* has 2 *letters*) 0 or 1. P(0) = 0.25, P(1) = 0.75.

H = 0.25 * log_2 (1/0.25) +

0.75 * log_2(1/0.75)

= 0.8113 bits

We need at least 0.8113 bits per symbol in encoding.

The Entropy of an Image

- An grayscale image with 256 possible levels. A={0, 1, 2, …, 255}. Assuming the pixels are independent and the grayscales are have equal probabilities,

  H = 256 * 1/256 *log2(1/(1/256)) = 8bits

- What about an image with only 2 levels 0 and 255? Assuming, P(0) = 0.5 and P(255) = 0.5.

  H = 1 bit

Estimate the Entropy

Assuming the symbols are independent:

a a a b b b b c c c c d d

$P(a) = 3/13$
$P(b) = 4/13$
$P(c) = 4/13$
$P(d) = 2/13$

$$H = [-P(a)\log_2 P(a)] + [-P(b)\log_2 P(b)] + [-P(c)\log_2 P(c)] + [-P(d)\log_2 P(d)]$$
$$= 1.95\text{bits}$$

Entropy encoding schemes

- Huffman Encoding

- Shannon Fano

Huffman Encoding

Huffman Encoding

- Statistical encoding
- Entropy encoding
- Starts with histogram then normalized histogram
- Recall normalized histogram numbers are probabilities
- To determine Huffman code, it is useful to construct a binary tree
- Leaves are characters to be encoded
- Nodes carry occurrence probabilities of the characters belonging to the sub-tree

Huffman Encoding (Example)

Step 1 : Sort all Symbols according to their probabilities (left to right) from Smallest to largest

these are the leaves of the Huffman tree

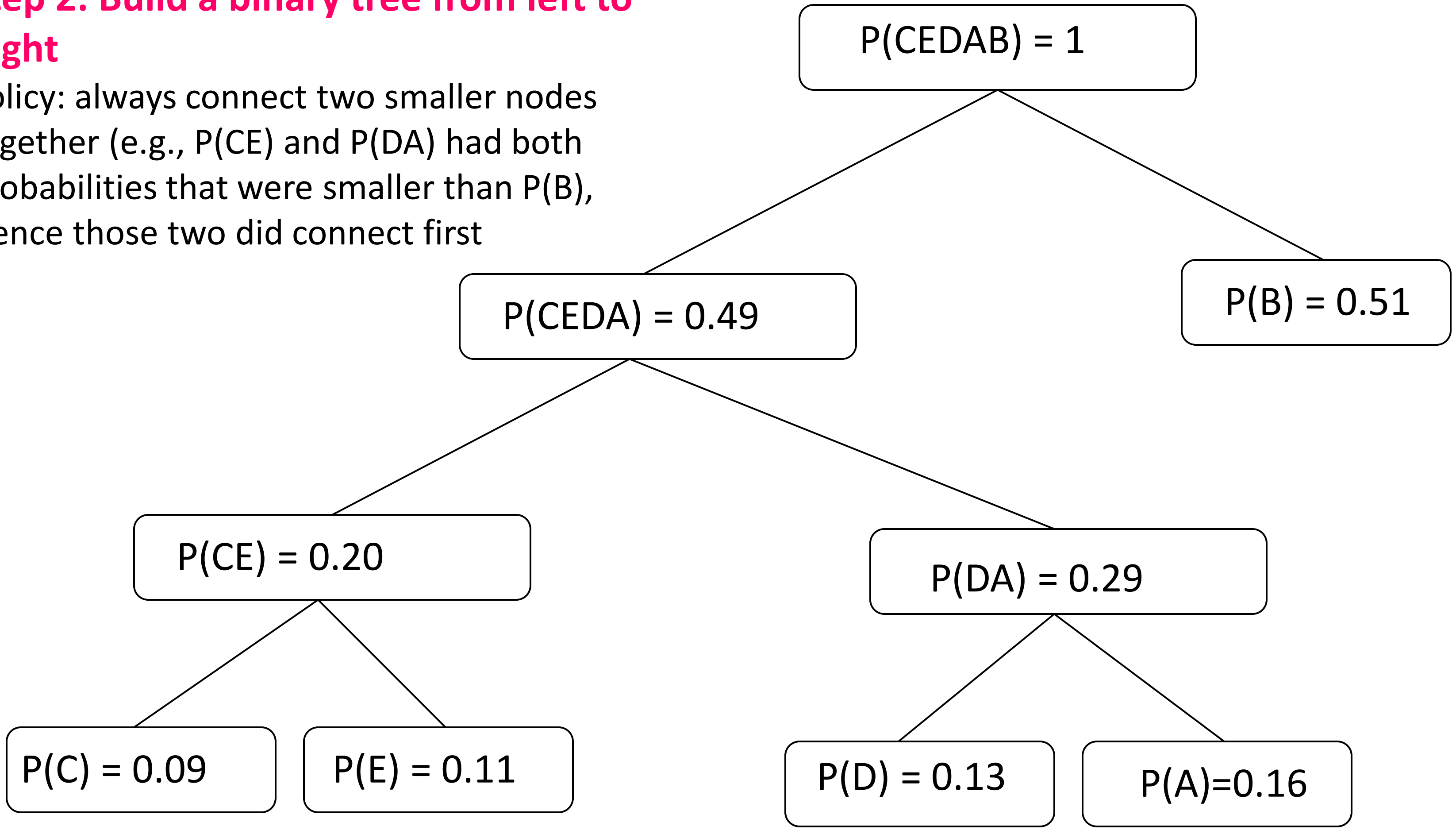
- Say we have symbols A,B,C,D,E with probabilities 0.16,0.51,0.09,0.13,0.11 in sequence
- Symbols could represent color codes, gray levels, audio sound level

P(C) = 0.09	P(E) = 0.11	P(D) = 0.13	P(A)=0.16	P(B) = 0.51

Huffman Encoding (Example)

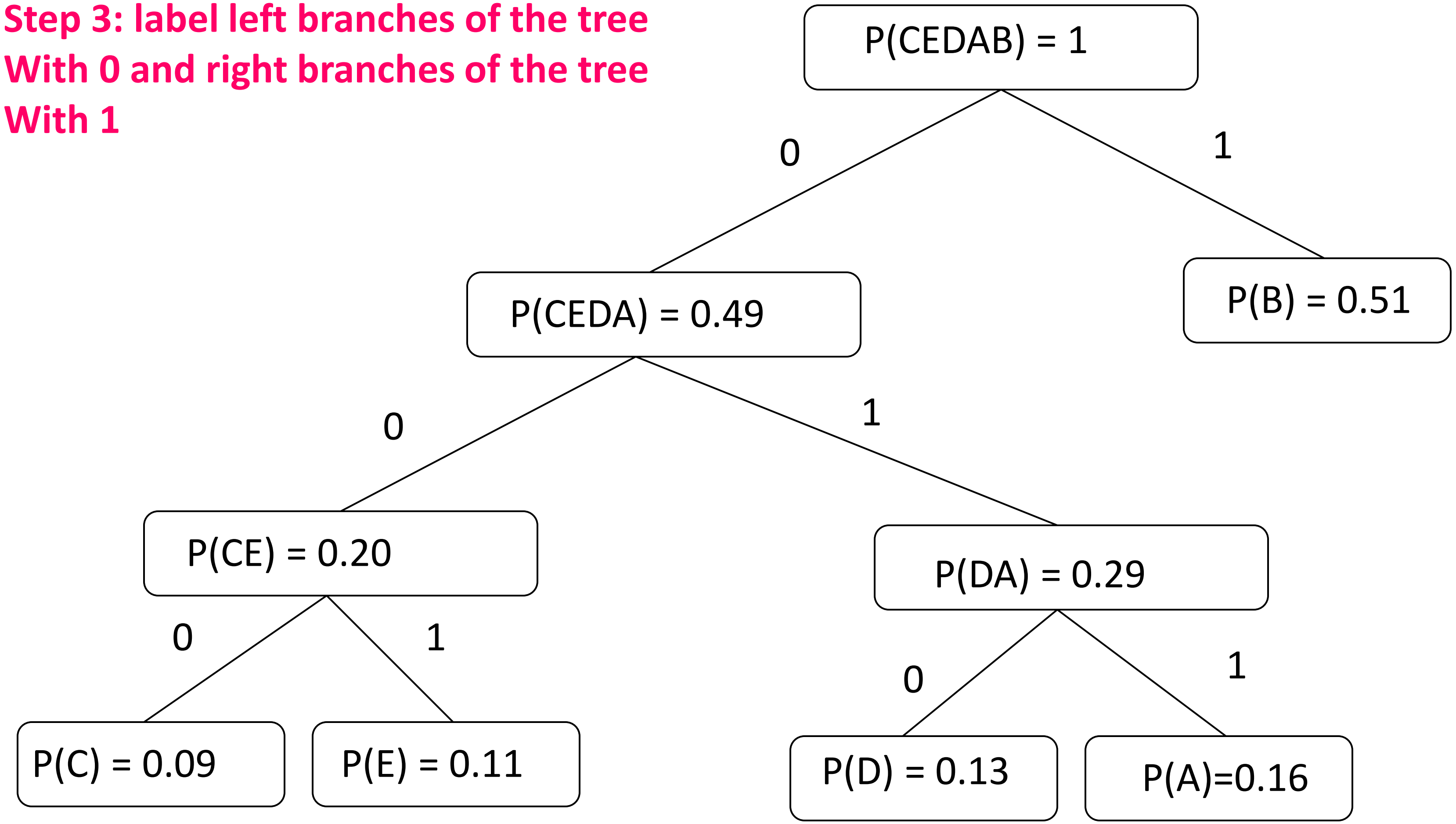
Step 2: Build a binary tree from left to Right

Policy: always connect two smaller nodes together (e.g., P(CE) and P(DA) had both Probabilities that were smaller than P(B), Hence those two did connect first



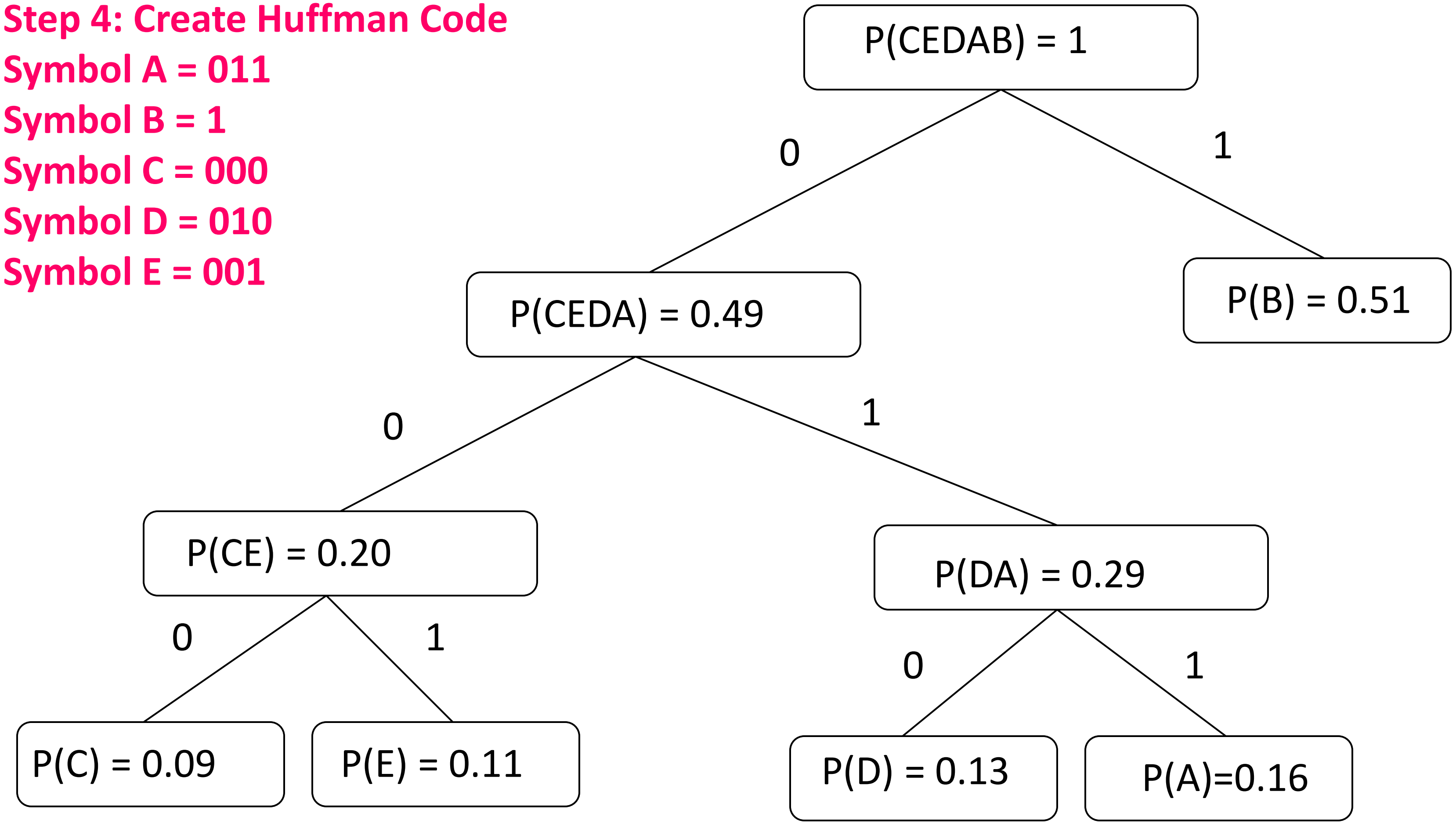
Huffman Encoding (Example)

Step 3: label left branches of the tree With 0 and right branches of the tree With 1



Huffman Encoding (Example)

Step 4: Create Huffman Code
Symbol A = 011
Symbol B = 1
Symbol C = 000
Symbol D = 010
Symbol E = 001



Huffman Encoding table

- **A = 011**
- **B = 1**
- **C = 000**
- **D = 010**
- **E = 001**

- Table needed for encoding and decoding process
- Table to be kept in the encoded file for the other side to be able decode
- No code is a prefix of other code
- Sequence of bits are uniquely interpreted by specific table

AAABCCDDEAAA
Encoded binary output is :
 011011011100000010010001011011011   encoded data

Huffman Encoding table

- **A = 011**
- **B = 1**
- **C = 000**
- **D = 010**
- **E = 001**

- ➢ Table needed for encoding and decoding process
- ➢ Table to be kept in the encoded file for the other side to be able decode
- ➢ No code is a prefix of other code
- ➢ Sequence of bits are uniquely interpreted by specific table

AAABCCDDEAAA
Encoded binary output is :
011 011 011 1 000 000 010 010 001 011 011 011 encoded data
AAABCCDDEAAA DECODED FILE

Huffman Encoding table

- **A = 011**
- **B = 1**
- **C = 000**
- **D = 010**
- **E = 001**

- Table needed for encoding and decoding process
- Table to be kept in the encoded file for the other side to be able decode
- No code is a prefix of other code
- Sequence of bits are uniquely interpreted by specific table

AAABCCDDEAAA
Encoded binary output is :
011011011100000001001001011011011 encoded data
AAABC.....

ABCDE
Size before compression=5*8=40
Size after compression=3+1+3+3+3=13
CR=40/13

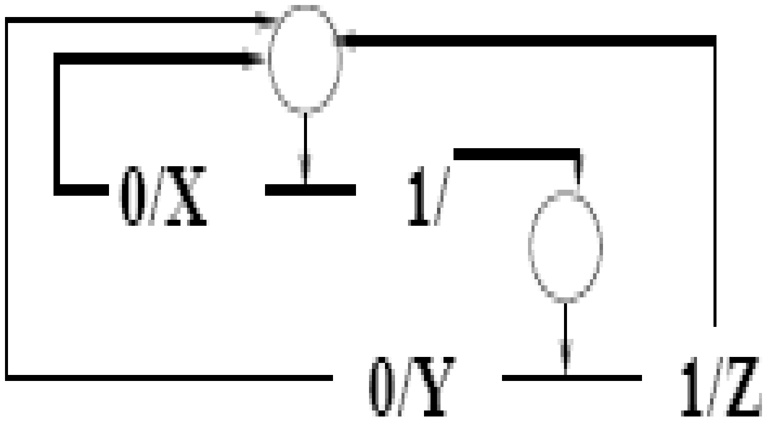
Huffman Decoding

- Assume Huffman Table
- Symbol Code

Symbol	Code
X	0
Y	10
Z	11

Consider encoded bitstream: 000101011001110

xxxyyzxxzy



What is the decoded string?

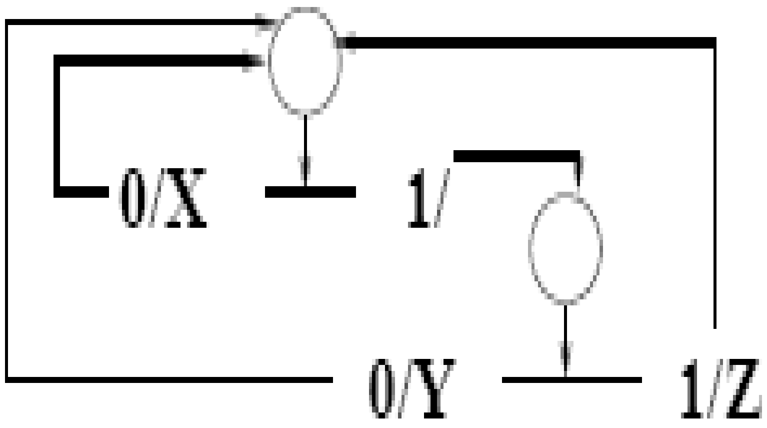
Huffman Decoding

- Assume Huffman Table
- Symbol Code

X	0
Y	10
Z	11

Consider encoded bitstream: 000101011001110

xxxyyzxxzy



What is the decoded string?

Huffman Example

- Construct the Huffman coding tree (in class)
- General Huffman rules
- If P(x)>P(y) then
- Code length(x)<= code length(y)

Home work: design binary Huffman tree

Symbol (S)	*P*(S)
A	0.25
B	0.30
C	0.12
D	0.15
E	0.18

Characteristics of Solution

Symbol (S)	Code
A	01
B	11
C	100
D	101
E	00

Example Encoding/Decoding

Encode "BEAD"

$\Rightarrow$110001101

$\Rightarrow$Decode "0101100"

Symbol (S)	Code
A	01
B	11
C	100
D	101
E	00

Entropy (Theoretical Limit)

$$H = \sum_{i=1}^{N} -p(s_i)\log_2 p(s_i)$$

$= -.25 * \log_2 .25 +$
$\quad -.30 * \log_2 .30 +$
$\quad -.12 * \log_2 .12 +$
$\quad -.15 * \log_2 .15 +$
$\quad -.18 * \log_2 .18$

H = 2.24 bits

Symbol	*P*(S)	Code
A	0.25	01
B	0.30	11
C	0.12	100
D	0.15	101
E	0.18	00

Average Codeword Length

$$L = \sum_{i=1}^{N} p(s_i) codelength(s_i)$$

= .25(2) +
.30(2) +
.12(3) +
.15(3) +
.18(2)

L = 2.27 bits

Symbol	P(S)	Code
A	0.25	01
B	0.30	11
C	0.12	100
D	0.15	101
E	0.18	00

Huffman encoding

- Entropy encoding
- One of Best entropy encoding
- Exact
- Variable length
- Encoding, Complexity ?
- Must send code book with the data
- Must determine frequency distribution

Home work 1

- Compute Entropy (H)
- Build Huffman tree
- Compute average code length
- Compute compression ratio assume original 8 bit
- Code “BCCADE”
- Try to Decode ‘0110100110’ . Is valid Huffman code?

Symbol (S)	*P*(S)
A	0.1
B	0.2
C	0.4
D	0.2
E	0.1

Shannon Fano entropy encoding

- Create a list of probabilities or frequency counts for the given set of symbols so that the relative frequency of occurrence of each symbol is known.

- Sort the list of symbols in decreasing order of probability, the most probable ones to the left and least probable to the right.

- Split the list into two parts, with the total probability of both the parts being as close to each other as possible.

- Assign the value 0 to the left part and 1 to the right part.

- Repeat the steps 3 and 4 for each part, until all the symbols are split into individual subgroups.

Shannon Fano entropy encoding

- Example for the following histogram
- A 100
- B 200
- C 400
- D 800
- E 1000
- F 100
- G 400

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000
- B 200 800
- C 400 400
- D 800 400
- E 1000 200
- F 100 100
- G 400 100

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1
- B 200 800 1

- C 400 400 0
- D 800 400 0
- E 1000 200 0
- F 100 100 0
- G 400 100 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0

- C 400 400 0
- D 800 400 0
- E 1000 200 0
- F 100 100 0
- G 400 100 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0
- --
- C 400 400 0
- D 800 400 0

- E 1000 200 0
- F 100 100 0
- G 400 100 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0

- C 400 400 0 1
- D 800 400 0 1

- E 1000 200 0 0
- F 100 100 0 0
- G 400 100 0 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0

- C 400 400 0 1 1
- D 800 400 0 1 0

- E 1000 200 0 0
- F 100 100 0 0
- G 400 100 0 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0

- C 400 400 0 1 1
- D 800 400 0 1 0

- E 1000 200 0 0

- F 100 100 0 0
- G 400 100 0 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0

- C 400 400 0 1 1
- D 800 400 0 1 0

- E 1000 200 0 0 1

- F 100 100 0 0 0
- G 400 100 0 0 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0

- C 400 400 0 1 1
- D 800 400 0 1 0

- E 1000 200 0 0 1

- F 100 100 0 0 0 1
- G 400 100 0 0 0 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0
- ---
- C 400 400 0 1 1
- D 800 400 0 1 0
- --
- E 1000 200 0 0 1
- ---------------------------------------
- F 100 100 0 0 0 1
- G 400 100 0 0 0 0

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0

- C 400 400 0 1 1
- D 800 400 0 1 0

- E 1000 200 0 0 1

- F 100 100 0 0 0 1
- G 400 100 0 0 0 0

Shannon Fano encoding table:

Symbol	code	length
A	0001	4
B	001	3
C	011	3
D	10	2
E	11	2
F	0000	4
G	010	3

Shannon Fano entropy encoding

- Example for the following histogram
- A 100 1000 1 1
- B 200 800 1 0
- ---
- C 400 400 0 1 1
- D 800 400 0 1 0
- ---
- E 1000 200 0 0 1
- -----------------------------------
- F 100 100 0 0 0 1
- G 400 100 0 0 0 0

Encoding AAABBFFGD
00010001000100100100000000001010
Decode: 0001,11,010,10,11,11,011, ----- valid encoded file
00011101010111110110 invalid encoded file
AEGDEEC

A	0001	4
B	001	3
C	011	3
D	10	2
E	11	2
F	0000	4
G	010	3

Avr lngth= 4 *100/3000+3*200/3000+3*400/3000 +2*800/3000+2*1000/3000+4*100/300+3*400/300

H=Sum(prob *log2 (1/prob)

CR=(8*3000)/(4*100+3*200+3*400+2 *800+2*1000+4*100+3*400}