ECE462 – Lecture 9

• Information Theory Fundamentals/Entropy

Given an information source with alphabet

$$S = \{s_1, s_2, s_3 \dots \dots, s_n\}$$

and $P_i = P[S = S_i]$

The "uncertainty" or "self-information" of the event $\{S = Si\}$ is given by

$$\log_2 \frac{1}{P_i} = - \log_2 P_i$$

The "entropy" of the source is defined as

η = H(S) = $\sum_i P_i \log_2 \frac{1}{P_i}$ = - $\sum_{i=1}^{n} P_i \log_2 P_i$ (bits)

- **The entropy is a measure of disorder**

- The higher the order the less the entropy
- The higher the randomness of the source the higher the entropy

In Coding Theory

1. The average codeword length of any code (in bits) cannot be less than the entropy
2. The entropy of a source ***S*** represents the minimum average number of bits required to establish the “outcome” of ***S***
3. In lossless compression the lower possible “average” bit representation of a source is given by the entropy of the source

- **Examples:**

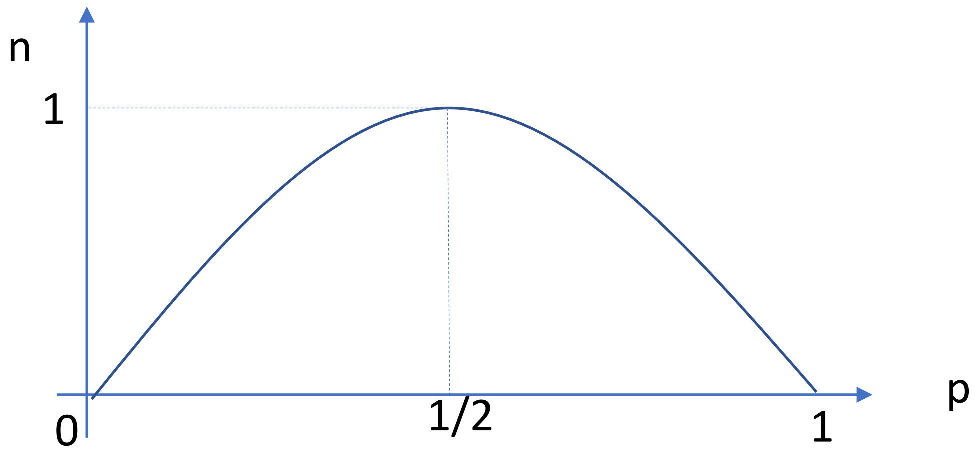
1. Consider an image (grey scale) 256 possible levels each having the same chance of occurring. Then

   $P_i = P(S = Si) = \frac{1}{256}$ i=0,….,255

   Entropy = n = H(S) = $\sum_{i=0}^{255} \frac{1}{256} \log_2 256$ (8 bits)

2. Consider a black & white image with P[S = black] = P, P[S=white] = 1-P

   Then n=H(S) = -P $\log_2 P - (1 - P) \log_2 (1 - P) \leq$ 1bit



- **Example:**

A basket contains 16 balls.

4 balls are labelled as “A”

4 balls are labelled as “B”

2 balls are labelled as “C”

2 balls are labelled as “D”

Remaining 4 balls are labelled as “E”,”F”,”G”,”H”

D.H picks a ball at random and records the label.

Discuss what strategies a student in ECE462 can use to find out the label of the ball through a series of yes/no questions. Compare the average number of questions asked to the entropy of the information source given here by the set ***S*** = {A,B,C,D,E,F,G,H}

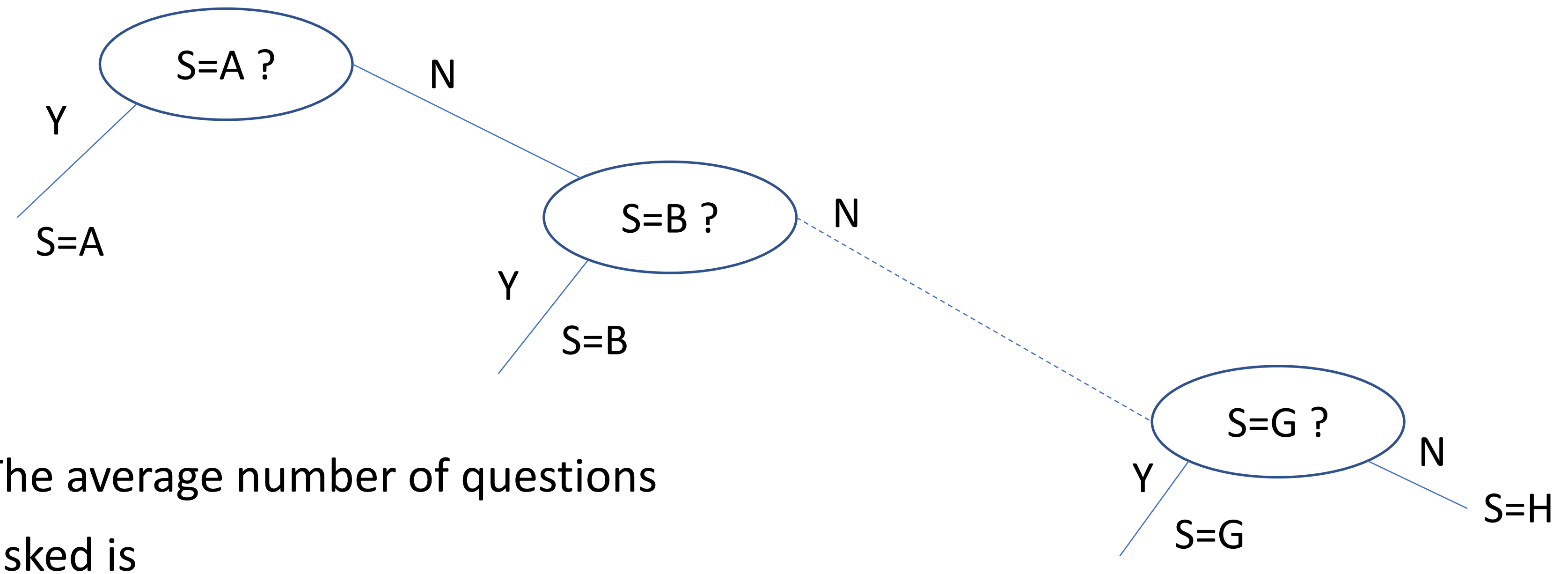
Obviously: $P\{A\} = P\{B\} = \frac{4}{16} = \frac{1}{4}$

$P\{C\} = P\{D\} = \frac{2}{16} = \frac{1}{8}$

$P\{E\} = P\{F\} = P\{G\} = P\{H\} = \frac{1}{16}$

- **<u>Strategy1:</u>** Starting wit the highest probability events ask the question {is S = "   "} sequentially until the answer is "Yes".

The following tree is constructed



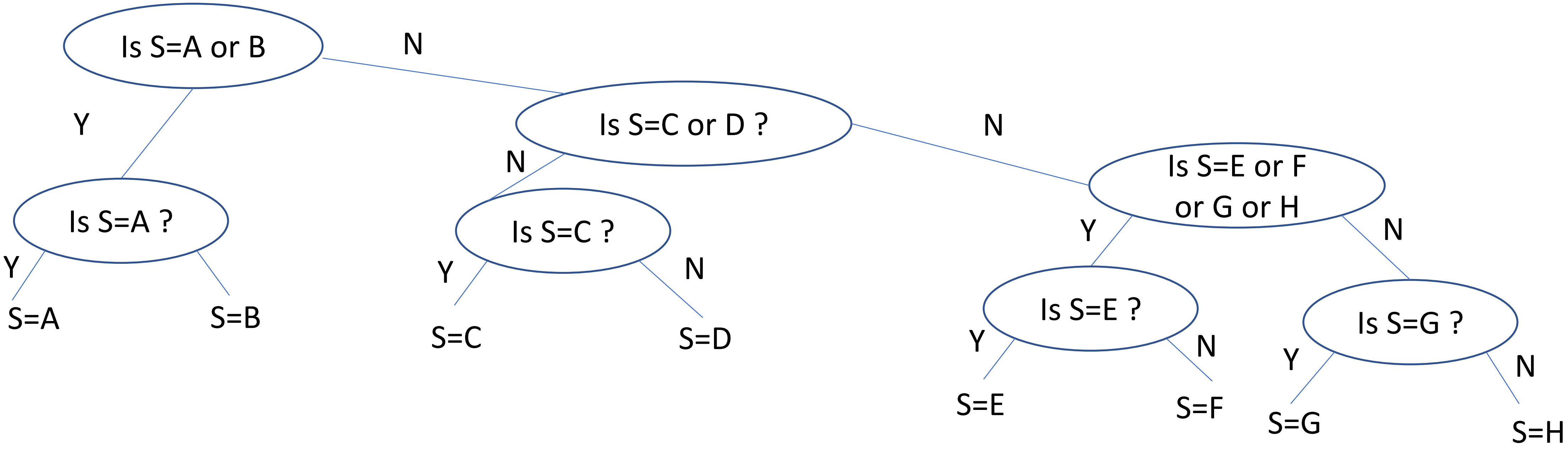
The average number of questions

asked is

1 quest. for AxP(S=A) + 2 quest. for Bx P(S=B) + 3 quest. for Cx P(S=C)+ 4 quest. for Dx P(S=D) + 5 quest. for Ex P(S=E)+ 6 quest. for Fx P(S=F) + 7 quest. for G or H.xP(S=G or S=H)

$$=1\frac{1}{4}+2\frac{1}{4}+3\ \frac{1}{8}+4\frac{1}{8}+5\frac{1}{16}+6\ \frac{1}{16}+\ \frac{7}{16}+\frac{7}{16}=\ \frac{51}{16}$$

- **Strategy2:** Design the tree so that the answer yes/no are equiprobable.



Now we have:

asked is

2 quest. for Ax P(S=A) + 2 quest. for Bx P(S=B) + 3.quest. for Cx P(S=C)+ 3 quest. for D. xP(S=D) + 4 quest. for E or F or G or Hx P(S=E or F or G or H)

$=2\frac{1}{4}+2\frac{1}{4}+3\ \frac{1}{8}+3\ \frac{1}{8}+4.4\frac{1}{16}=\ \ \frac{44}{16}$

Now the entropy of the source is

$\eta = H(S) = -\frac{1}{4}\log_2\frac{1}{4} - \ldots\ldots\ldots\ldots - \frac{1}{16}\log_2\frac{1}{16} = \frac{44}{16}$

Thus, the second strategy achieves the entropy of the source.

- If we replace the yes with “1” and No by “0” we can use the following representation for each letter in the basket

Strategy 1

Count			bits
4	A	1	4
4	B	0 1	8
2	C	0 0 1	6
2	D	0 0 0 1	8
1	E	0 0 0 0 1	5
1	F	0 0 0 0 0 1	6
1	G	0 0 0 0 0 0 1	7
1	H	0 0 0 0 0 0 0	7
-			-
16			51

Average bit required = 51/16

Strategy 2

		bits
A	1 1	8
B	1 0	8
C	0 1 1	6
D	0 1 0	6
E	0 0 1 1	4
F	0 0 1 0	4
G	0 0 0 1	4
H	0 0 0 0	4
		-
		44

Average bit required = 44/16

- **<u>HUFFMAN CODING:</u>**

Given an information source **S**

1. Put all the symbols of the **S** sorted according to their frequency counts in descending order

2(a) Pick the two symbols with the lowest frequency, form a Huffman subtree with these two symbols as child nodes and create a parent node with frequency count the summation of the child nodes frequency counts.

2(b) Insert the parent node in the list so that order is maintained and delete the children from the list

Repeat until no symbol is left

3. Assign a codeword for each leaf in the tree based on the path from the root (Assign a "0" for each leaf on the left and "1" for each leaf on the right or vice versa)

Note: Huffman code is a variable length code

- **Question:**

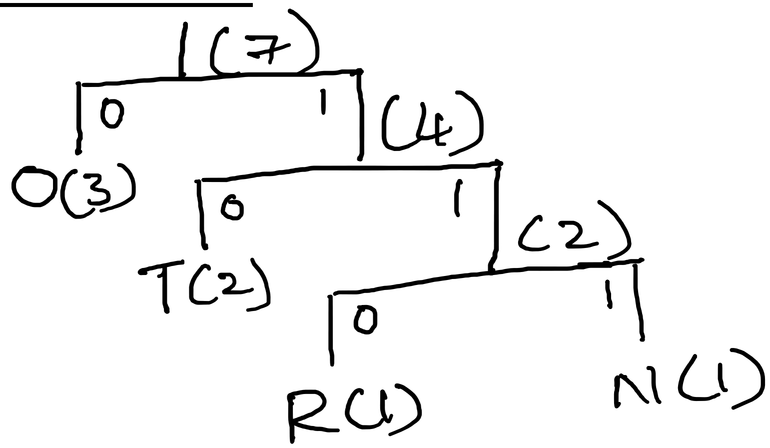
Is the previous example with the basket Huffman coded?

Example2: Consider coding the word “TORONTO”

Information Source: **S** = {O,T,R,N}

Frequency Count 3 2 1 1

Huffman Tree



symbol	count	code	#bits
O	3	0	3
T	2	10	4
R	1	110	3
N	1	111	3
	7		13

Average # of bits $= \frac{13}{7} = 1.8571 < 2$

- Without Huffman coding we need 2 bits per symbol to represent the 4 symbols
- Source entropy: $\frac{3}{7}\log_2\frac{7}{3} + \frac{2}{7}\log_2\frac{7}{2} + \frac{2}{7}\log_2 7 \ = 1.8424$

- **Important properties of Huffman coding:**

1. Unique prefix:

   No Huffman code is a prefix of another Huffman code

   Important property for efficient decoding

2. The higher the frequency of the symbol the shorter the Huffman code
3. In JPEG no Huffman code can consist of all 1-bits. In this case we can always insert a dummy symbol with frequency 0 which will assume this representation.

DECODING:

The obvious method to decode a Huffman code is to create a binary tree containing the values arranged according to their codes. Then start from the root using the value of bits read from the required data until a path is determined.

Student Name: Student Number:

University of Toronto
Faculty of Applied Science and Engineering

MIDTERM EXAMINATION
ECE462H1S, Multimedia Systems
Time allowed: 90 minutes
March 9, 2006
Examiner: D. Hatzinakos

Exam type A
Non-programmable Calculators are allowed

---

1) Suppose a data source produces ones and zeros independently with probabilities P(0)=7/12 and P(1)=5/12. What is the binary entropy of this source? (3 points)

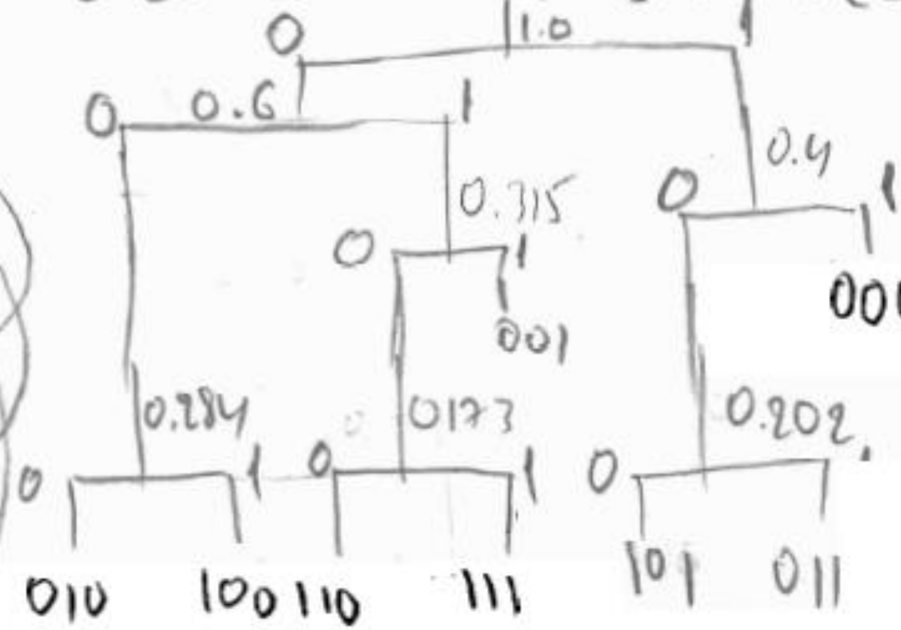
$$H = \frac{7}{12}\log_2\frac{12}{7} + \frac{5}{12}\log_2\frac{12}{5} = 0.9799 \text{ bits}$$

2) **Generate a Huffman code for blocks** of 3 bits (ones or zeros) from the source of question 1. What will be the codewords and the average codeword length for the code? (4 points)

First note that since 0 and 1 are produced independently then for a block of three bits, $P(b_1 b_2 b_3) = P(b_1)P(b_2)P(b_3)$

Then

	Probability
000	(0.1985)
001	(0.1418)
010	(0.1418)
011	(0.1013)
100	(0.1418)
101	(0.1013)
110	(0.1013)
111	(0.0723)



Codewords

000 → 11
001 → 011
010 → 000
011 → 101
100 → 001
101 → 100
110 → 0100
111 → 0101

Average code length $= 1 \cdot 2 \cdot 0.1985 + 3 \cdot 3 \cdot 0.1418 + 2 \cdot 3 \cdot 0.1013 + 1 \cdot 4 \cdot 0.1013 + 4 \cdot 0.0723 = 2.97$ bits