ENR-325/325L Principles of Digital Electronics and Laboratory

Xiang Li

Fall 2025

Coe College

Let's look into two of the coding method:

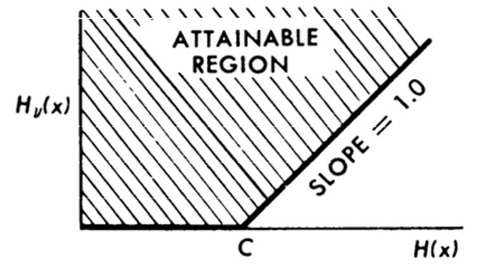
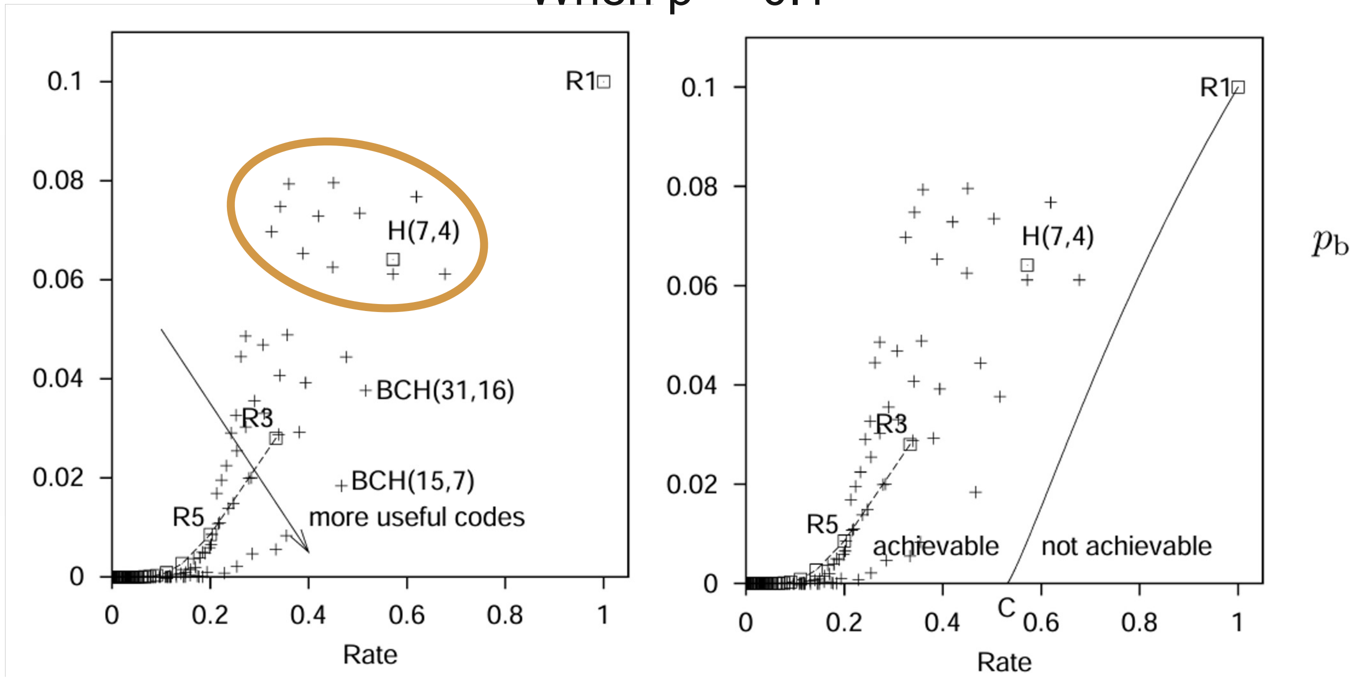


Fig. 9. — The equivocation possible for a given input entropy to a channel.

When p = 0.1



But how?

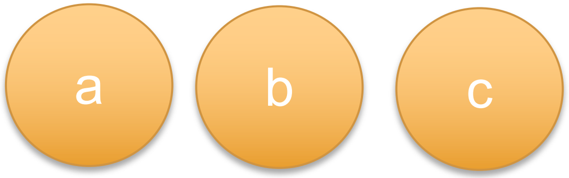
MacKay, David JC. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

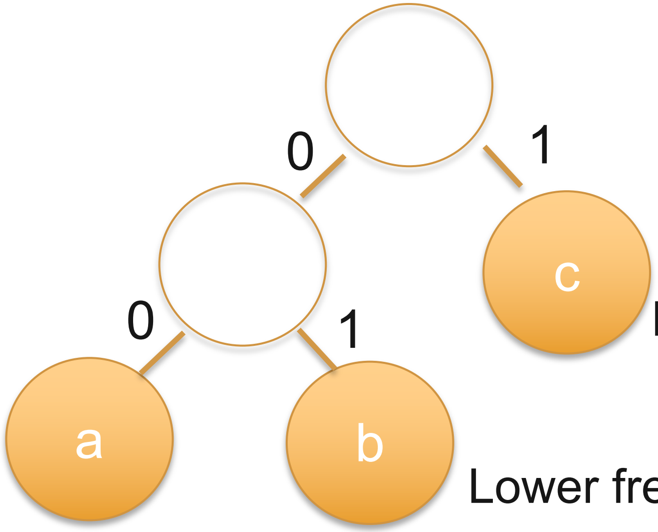
By smart encoding and decoding: Huffman coding

- Use a binary encoding system, assuming the decoder would have all the key as well.
- To use less bits: apply variable length based on the usage frequency.
- But the code has to be uniquely decipherable (prefix-free, many times just called prefix).

a = 1, b = 110, c = 10   Not prefix free! 110 can be “b” or “ac”
a = 0, b = 110, c = 10   Prefix free

Huffman tree (where each node has exactly two children)





Higher frequency: 1

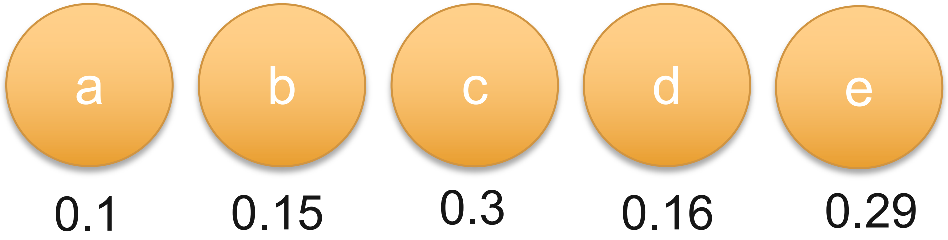
Lower frequency: 00

Lower frequency: 01

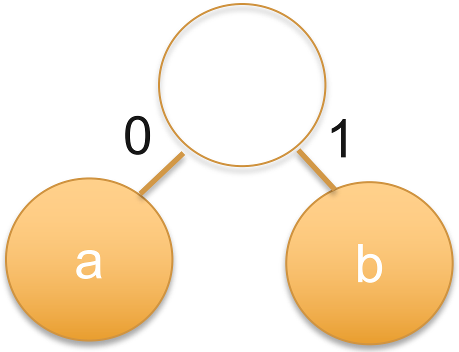
a = 00, b = 01, c = 1

COE COLLEGE

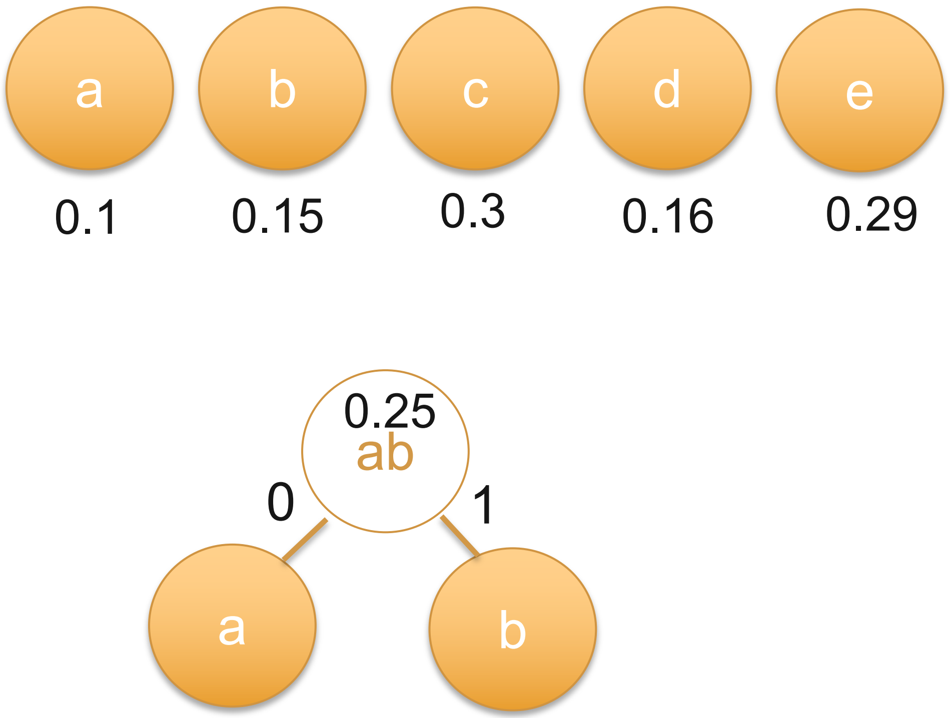
Sequence to construct Huffman tree



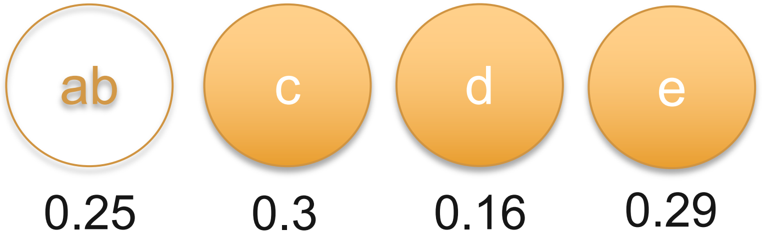
Pick two of the lowest frequency



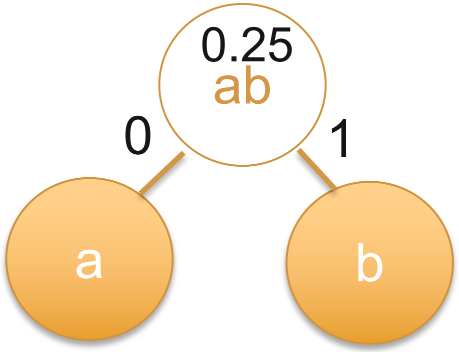
Sequence to construct Huffman tree

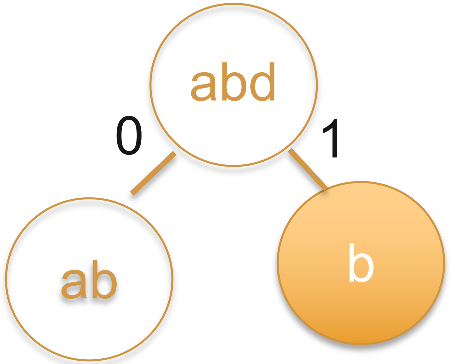


Sequence to construct Huffman tree

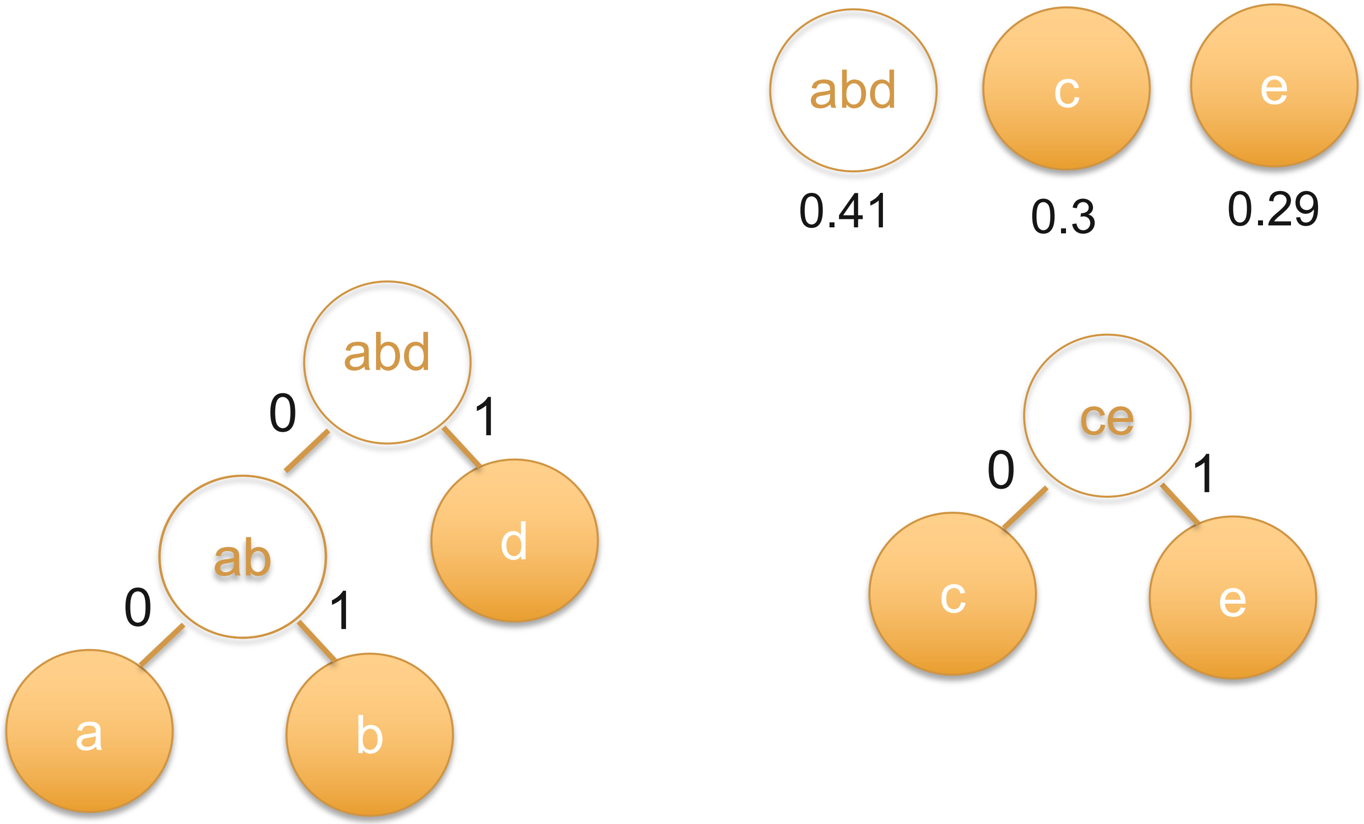


Pick two of the lowest frequency



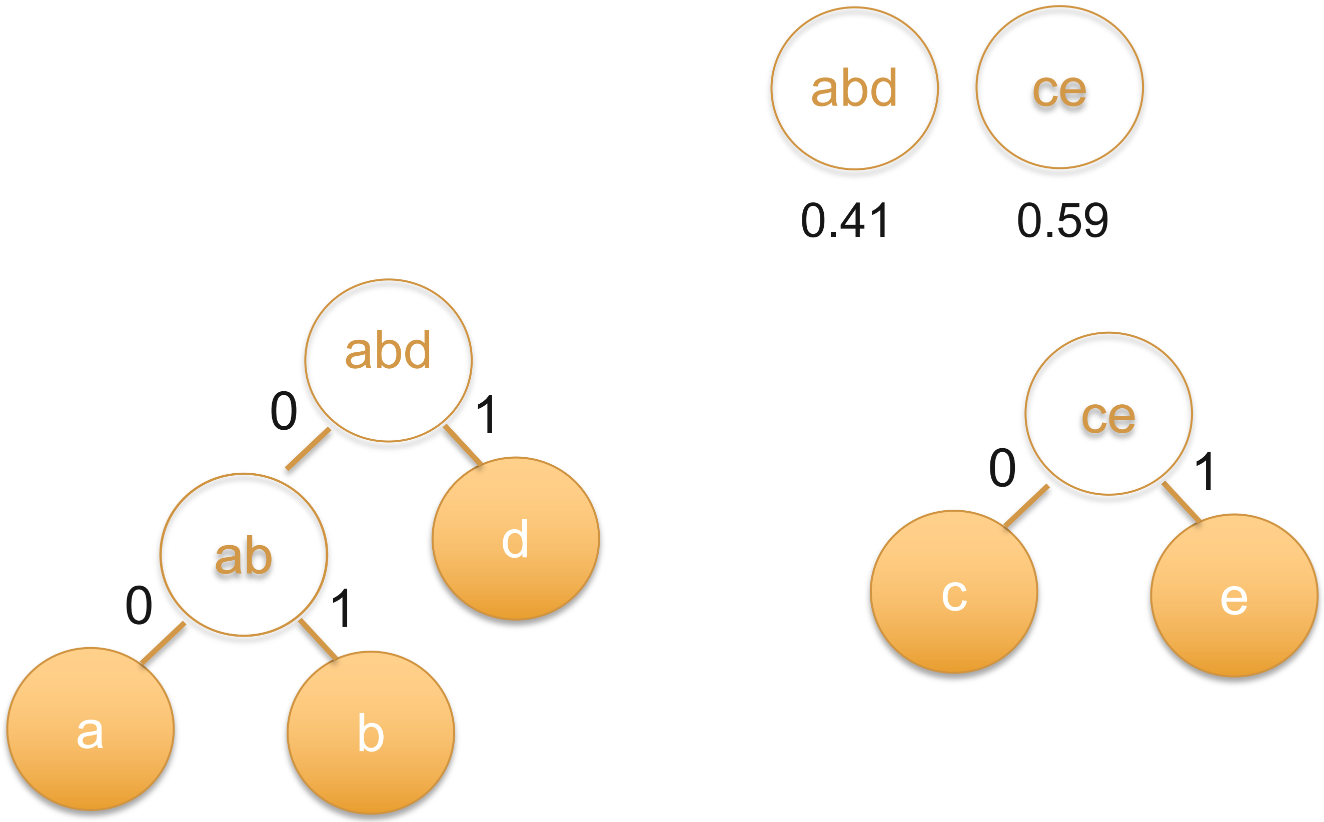


Sequence to construct Huffman tree



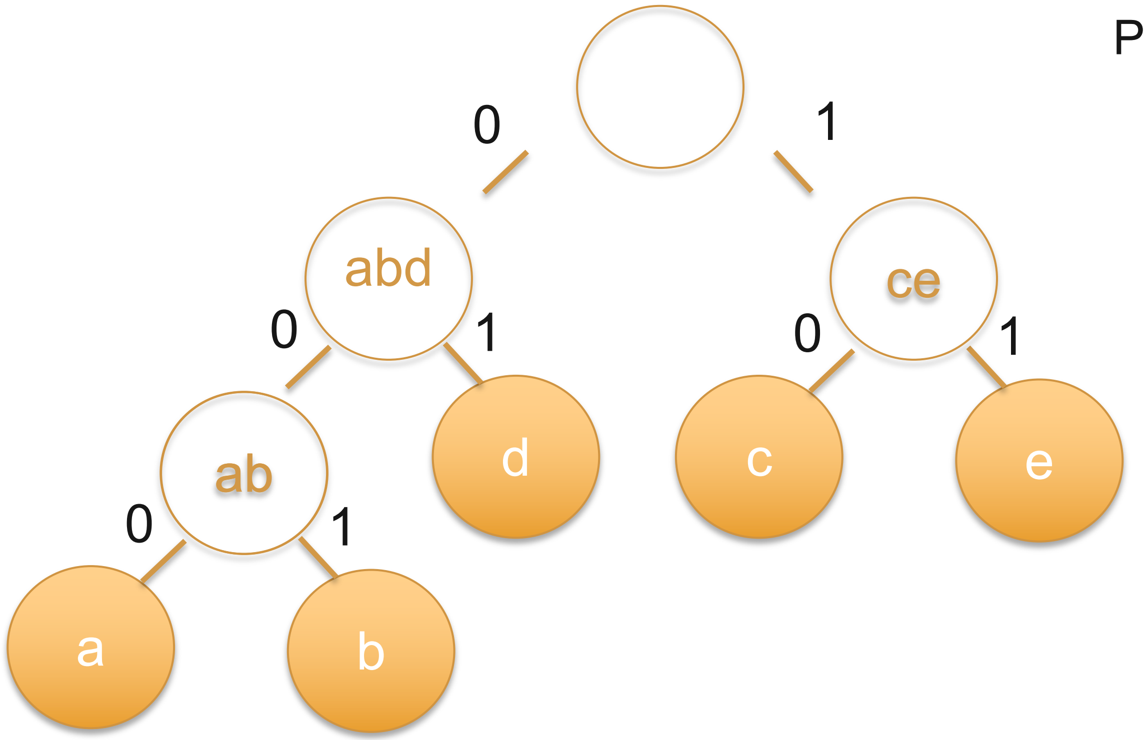
Pick two of the lowest frequency

Sequence to construct Huffman tree



Pick two of the lowest frequency

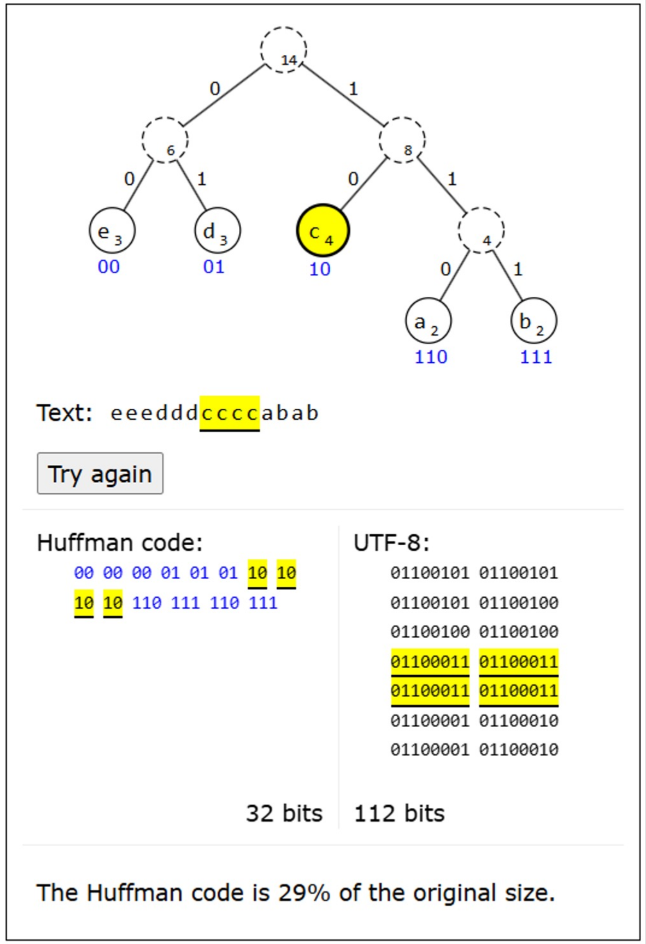
Sequence to construct Huffman tree



Pick two of the lowest frequency, done!

By smart encoding and decoding: Huffman coding

https://www.w3schools.com/dsa/dsa_ref_huffman_coding.php

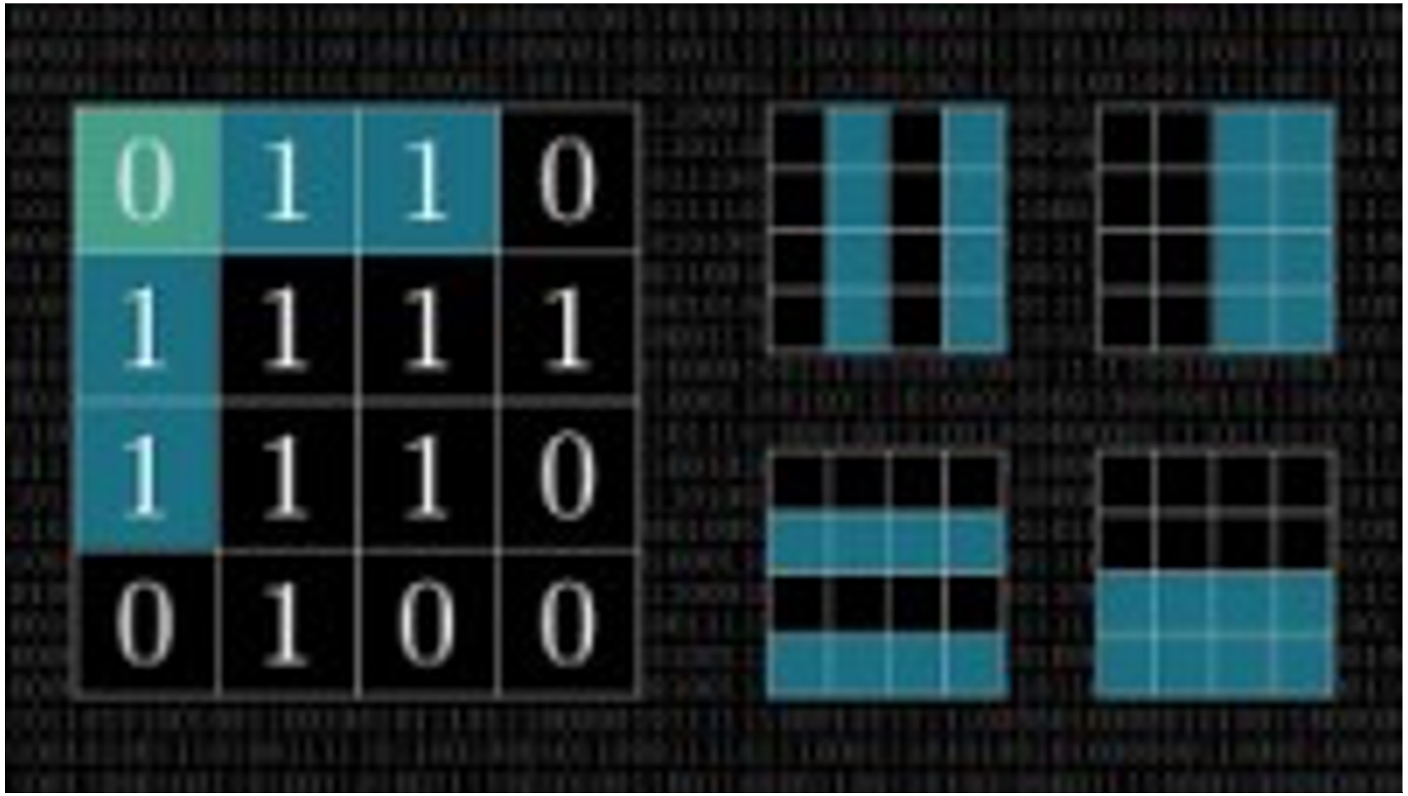


Input (*A*, *W*)	Symbol (a_i)	a	b	c	d	e	**Sum**
	Weights (w_i)	0.10	0.15	0.30	0.16	0.29	= 1
Output *C*	Codewords (c_i)	`010`	`011`	`11`	`00`	`10`	
	Codeword length (in bits) (ℓ_i)	3	3	2	2	2	
	Contribution to weighted path length ($\ell_i\, w_i$)	0.30	0.45	0.60	0.32	0.58	$L(C)$ = 2.25
Optimality	Probability budget ($2^{-\ell_i}$)	1/8	1/8	1/4	1/4	1/4	= 1.00
	Information content (in bits) ($-\log_2 w_i$) ≈	3.32	2.74	1.74	2.64	1.79	
	Contribution to entropy ($-w_i \log_2 w_i$)	0.332	0.411	0.521	0.423	0.518	$H(A)$ = 2.205

https://en.wikipedia.org/wiki/Huffman_coding

The hamming code

- Introduce redundancy, aka parity bits to check for errors.
- The receiver can use the parity bits to pinpoint and correct single-bit error.



https://www.youtube.com/watch?v=X8jsijhllIA&t=156s

Can also check:
https://harryli0088.github.io/hamming-code/

COE COLLEGE

The hamming code

- A bit more information to cover:

MONOGRAPH 1757

BELL TELEPHONE SYSTEM

TECHNICAL PUBLICATIONS

Error detecting and error correcting codes

by R. W. Hamming

We shall require that this checking number give the position of any single error, with the zero value meaning no error in the symbol. Thus the check number must describe $m + k + 1$ different things, so that

$$2^k \geq m + k + 1$$

TABLE I

n	m	Corresponding k
1	0	1
2	0	2
3	1	2
4	1	3
5	2	3
6	3	3
7	4	3
8	4	4
9	5	4
10	6	4
11	7	4
12	8	4
13	9	4
14	10	4
15	11	4
16	11	5
	Etc.	

2^8=256	256-8-1=247	8

Parity bits	Total bits	Data bits
m	$n = 2^m - 1$	$k = 2^m - m - 1$

https://en.wikipedia.org/wiki/Hamming_code