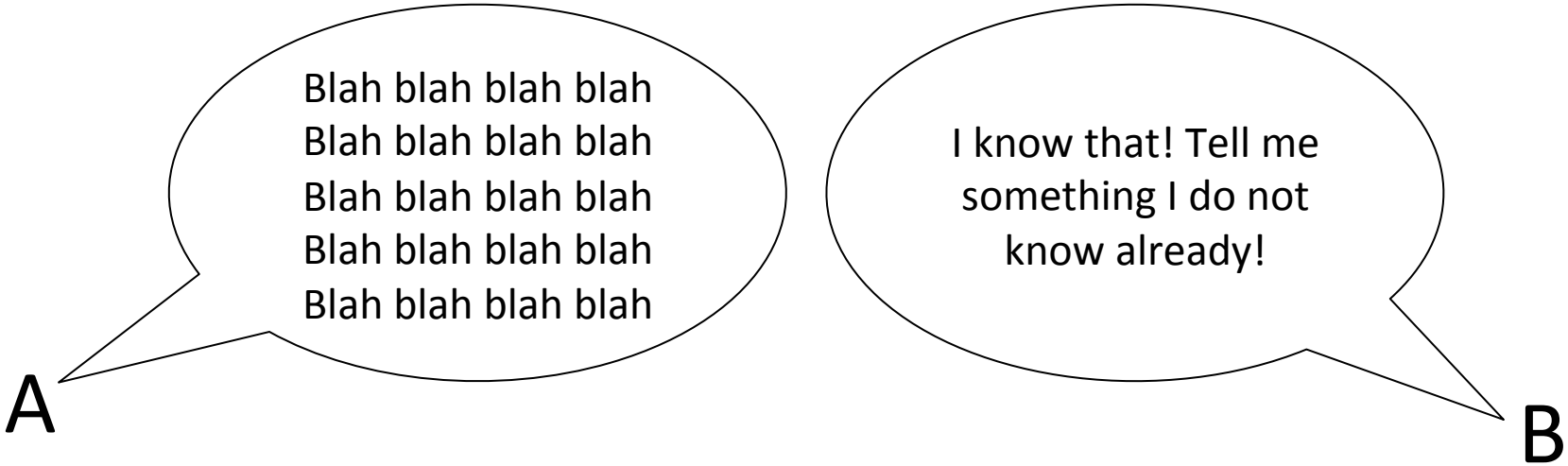
COMP250: Data compression

Jérôme Waldispühl
School of Computer Science
McGill University

Based on slides from M. Langer (McGill) and (goodrich & Tamassia, 2009)

Information Theory

- When A communicates a message to B, A sends a bit string that encodes the message.
- The amount of information in the message depends not of the number of bits sent, but rather on the probability of that message being sent.
- How much information does a message transmit?



Data compression

- When A communicates to B, they first agree on a code.
- They choose a code such that are the most likely to be sent are encoded using fewer bits. This yields shorter messages on average.
- The length of the message should be approximately equal to the amount of information communicated (Shannon, 1948).

Codes & Codewords

Suppose you have a sample space Σ (often called an alphabet).

Define a code to be a mapping :

$$C: \Sigma \rightarrow \{ \text{ bit string } \}$$

For any $x \in \Sigma$, $C(x)$ is the codeword of x.

The length of a codeword is the number of bits in that codeword.

Example:

$\Sigma = \{ A, C, G, T \}$

$C(A) = 00$, $C(C) = 01$, $C(G) = 10$, $C(T) = 11$

Extension of a code

For an code C on an alphabet Σ, we have a naturally defined code on sequences of elements from Σ.

Example:

C(AGAT) = C(A) C(G) C(A) C(T)

= 00 10 00 11

Note: We concatenate the codewords of the elements (letters) of the sequence.

Fixed length code

All codewords have the same length.

Example:

- $\Sigma = \{ A, C, G, T \}$

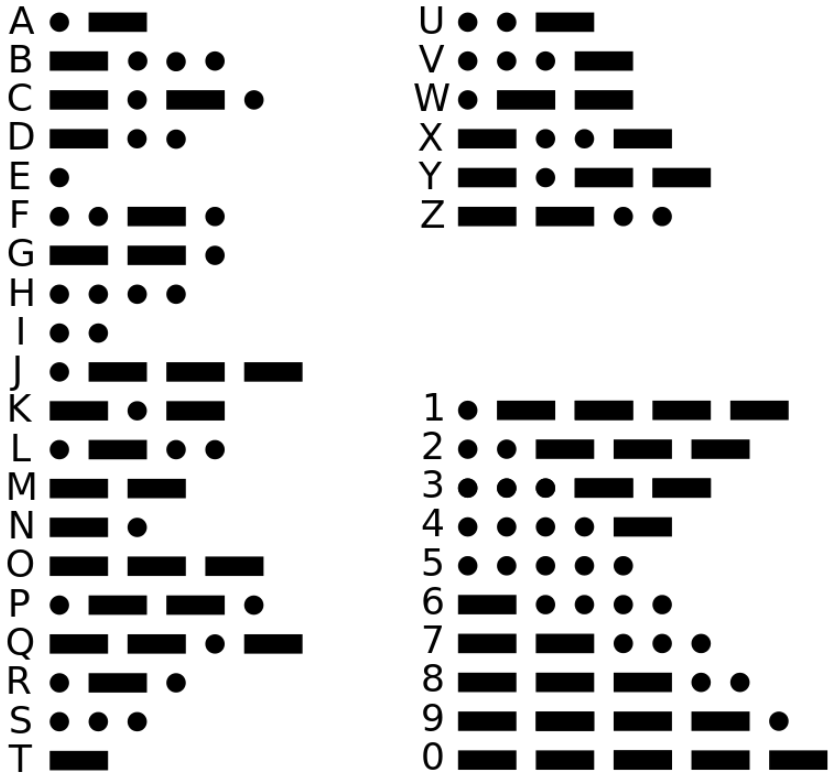
  $C(A) = 00, C(C) = 01, C(G) = 10, C(T) = 11$

- ASCII (8 bits), Unicode (16 bits)

Variable length code

Codewords can have different lengths.

Example: Morse code



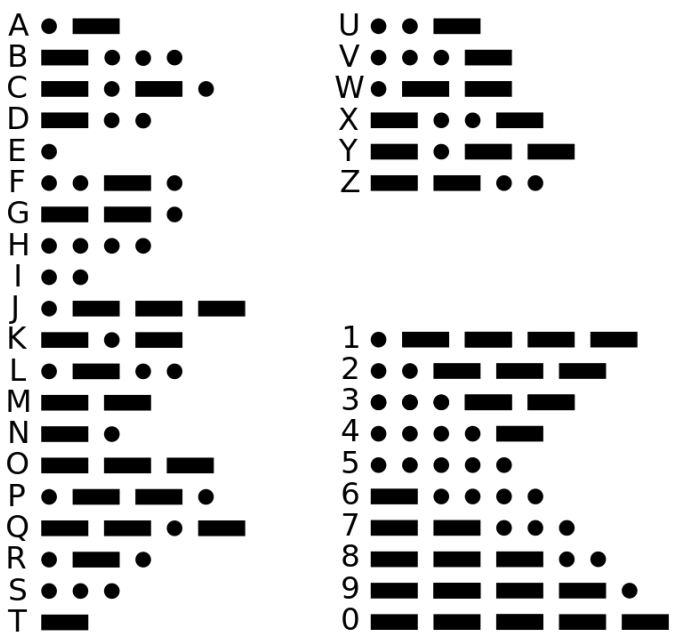
Note: More common letters have shorter codewords

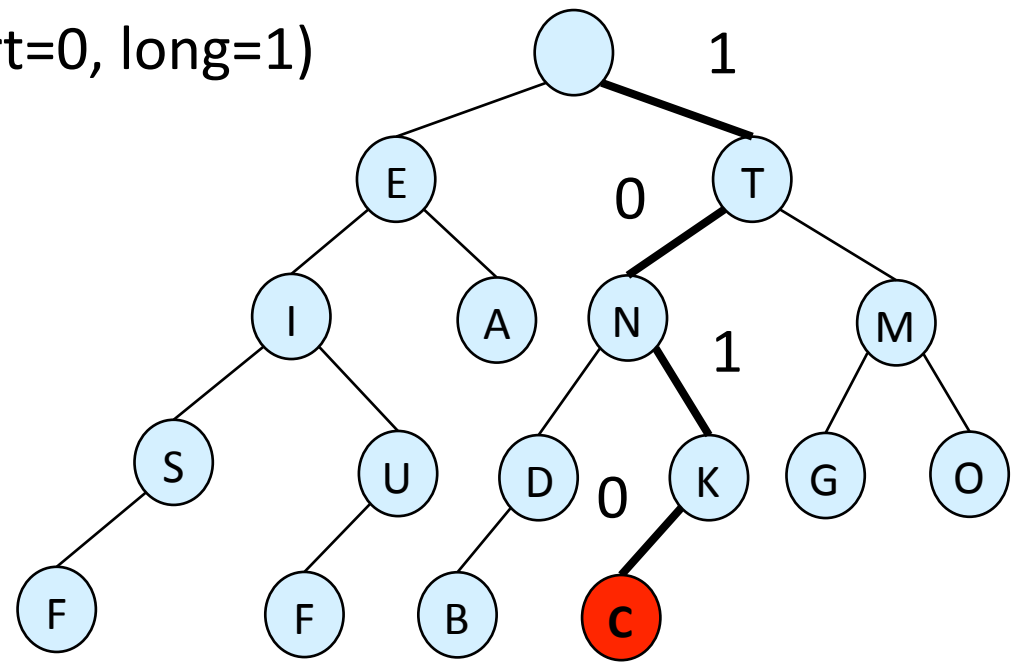
Tree representation

Any code can be represented by a binary tree. Each codeword is a path from the root to a node representing the element (letter) encoded.

Principle: 0 for left child, 1 for right child.

Example: Morse code (short=0, long=1)





C = 1010

Ambiguity

One big problem with Morse code is that messages are ambiguous.

Example:

C(A) = • — C(E)= • C(T) = —

- — = “A” or “ET” ??

Q: How to distinguish C(A) from C(ET) ?

A: Morse code inserts “space” between codewords… So it’s not really a binary code.

Prefix code

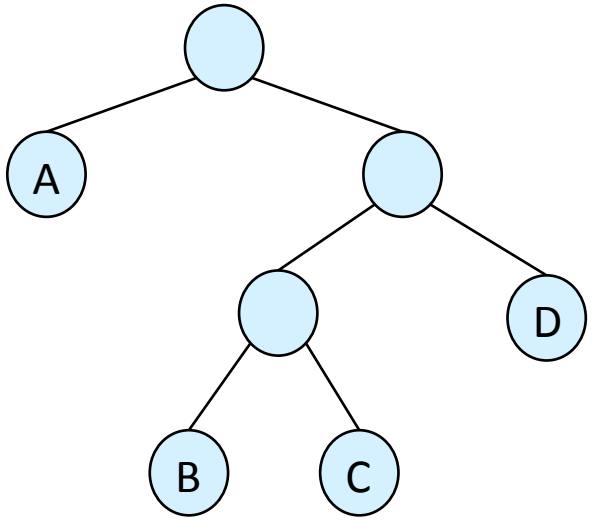
C is a prefix code if no codeword is a prefix of any other codeword.

Q: What does it mean in binary tree representations?

A: codewords are leaves!

Example:

C(A) = 0
C(B) = 100
C(C) = 101
C(D) = 11



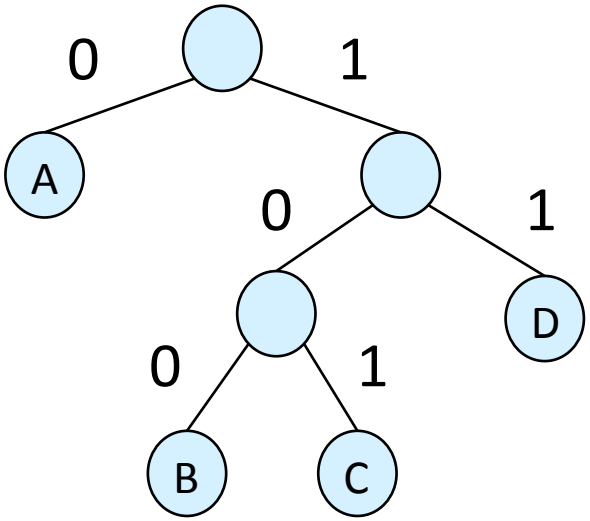
Desambiguation (Example)

Q: How to decode 0100110 ?

A:

0100110 A
0100110 AB
0100110 ABD
0100110 ABDA

Result: ABDA



Huffman's Algorithm

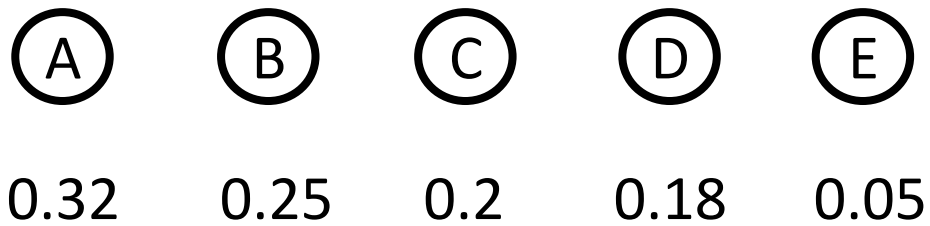
- Given a string X, Huffman's algorithm construct a prefix code that minimizes the size of the encoding of X
- It runs in time $O(n + d \log d)$, where n is the size of X and d is the number of distinct characters of X
- A heap-based priority queue is used as an auxiliary structure
- **Greedy algorithm!**

```
Algorithm HuffmanEncoding(X)
    Input string X of size n
    Output optimal encoding trie for X
    C ← distinctCharacters(X)
    computeFrequencies(C, X)
    Q ← new empty heap
    for all c ∈ C
        T ← new single-node tree storing c
        Q.insert(getFrequency(c), T)
    while Q.size() > 1
        f1 ← Q.minKey()
        T1 ← Q.removeMin()
        f2 ← Q.minKey()
        T2 ← Q.removeMin()
        T ← join(T1, T2)
        Q.insert(f1 + f2, T)
    return Q.removeMin()
```

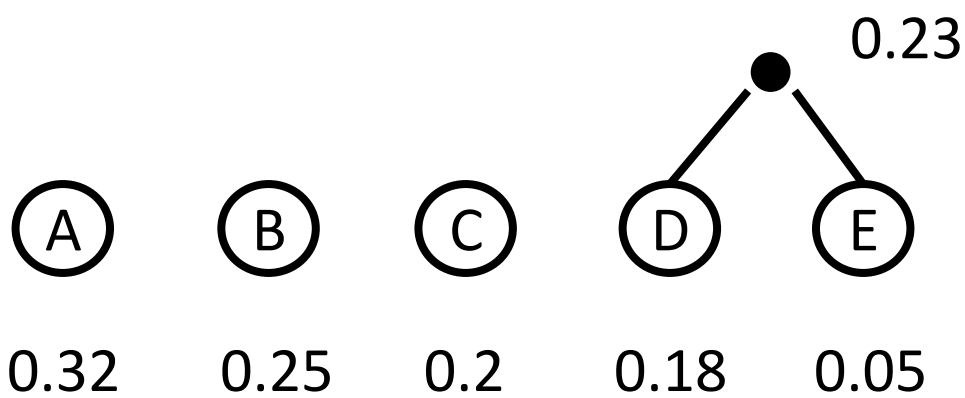
Example

Preprocessing: Compute the frequency of characters in string X.

P(A) = 0.32
P(B) = 0.25
P(C) = 0.2
P(D) = 0.18
P(E) = 0.05

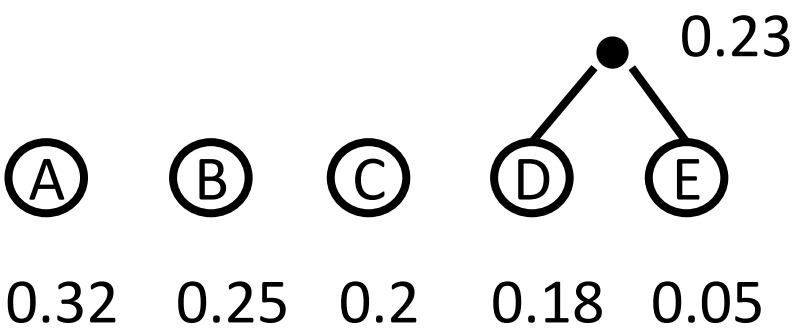


Merge two element with the lowest probabilities:



P(A) = 0.32
P(B) = 0.25
P(C) = 0.2
P({D,E}) = 0.23

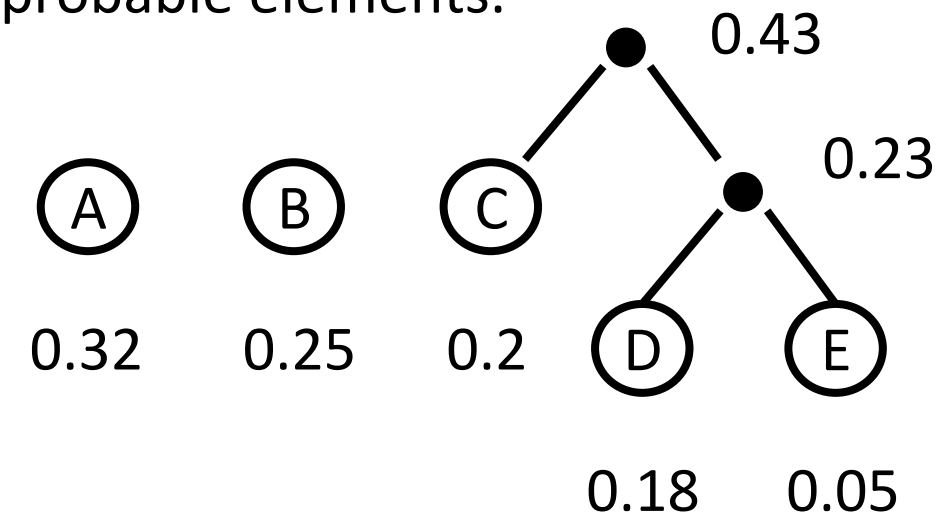
Example



P(A) = 0.32
P(B) = 0.25
P(C) = 0.2
P({D,E}) = 0.23

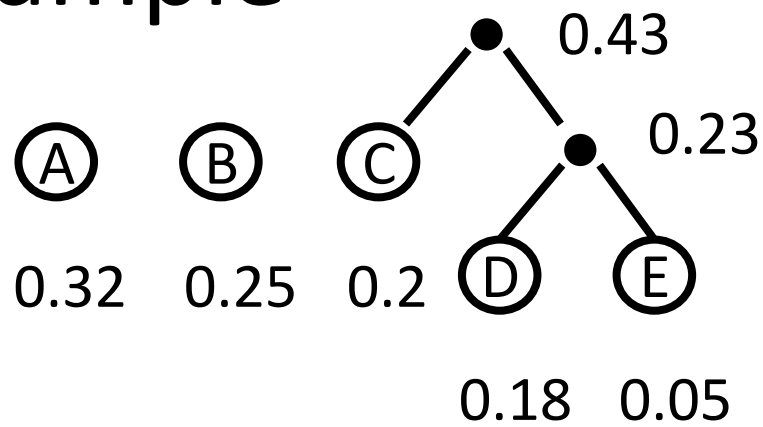
We have now 4 elements. The probability of this new element ({D,E}) is the sum of the probability of the elements merged. We merge the next two least probable elements.

P(A) = 0.32
P(B) = 0.25
P({C,D,E}) = 0.43

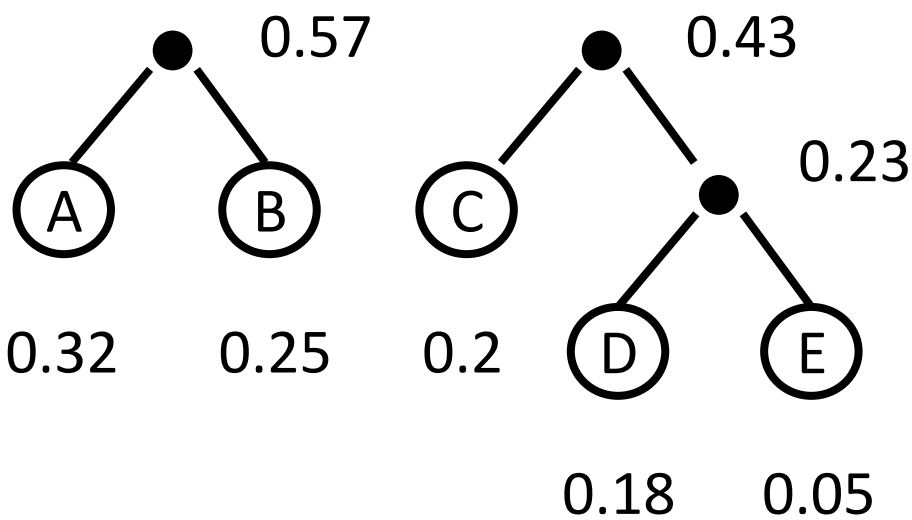


Example

P(A) = 0.32
P(B) = 0.25
P({C,D,E}) = 0.43



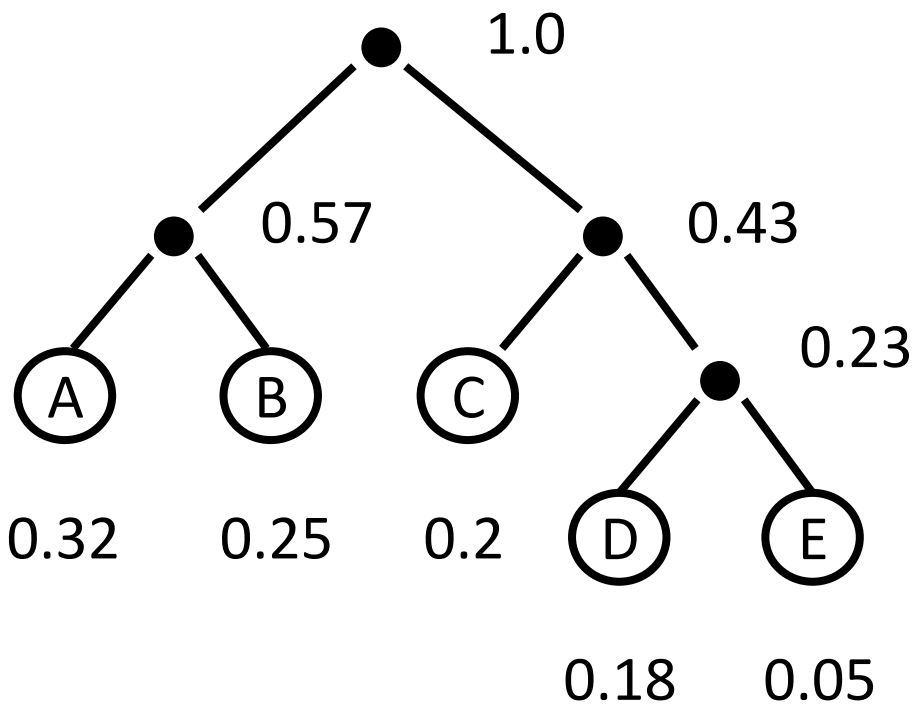
We have now 3 elements. The next two least probable elements are A and B, and we merge them again.



P({A,B}) = 0.57
P({C,D,E}) = 0.43

Example

Only two elements remain. We make these two elements siblings and we obtain an **optimal prefix code**!



Example (Encoding)

Original message

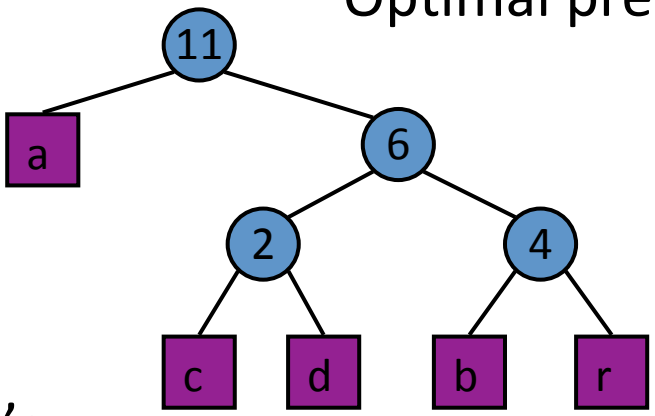
T = abracadabra

Frequencies

a	b	c	d	r
5	2	1	1	2

Huffman's algorithm

Optimal prefix code



Codewords

a : 0
b : 110
c : 100
d : 101
r : 111

Encode

Encoded message

T' = 0110111010001010110111 0

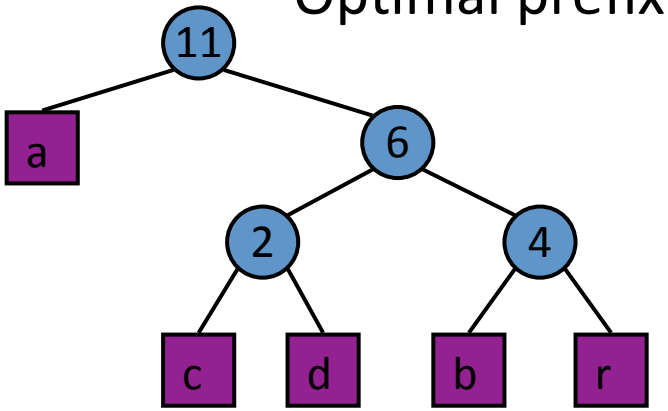
Example (Decoding)

Encoded message

M = 01101110100010101101110

+

Optimal prefix code



Decoded message

T = abracadabra

Example (Huffman)

T = abracadabra

Frequencies:

a	b	c	d	r
5	2	1	1	2