Principles of Data Communications

Reference Book: Data Communications and Networking by Behrouz A. Forouzan

Huffman Encoding

Text Compression

- Text files are usually stored by representing each character with an 8 bit ASCII code.
- ASCII encoding is an example of fixed length encoding, where each character is represented with the same number of bits.
- To reduce the space required to store a text file, we can exploit the fact that some characters are more likely to occur than others.
- Variable length encoding uses binary codes of different lengths for different characters; thus we can assign fewer bits to frequently used characters, and more bits to rarely used characters.

Example

- Text: “java”
- If fixed length encoding is used, atleast two bits per character are required here. Total 8 bits.
- Suppose- Variable length Encoding: a=“0”, j=“11”, v=“10”
- Encoded Text: 110100 (6 bits)

How to decode (problems in ambiguity)?

- Encoding: a=“0”; j=“01”; v=“00”
- Encoded text: 010000 (6 bits)
- Could be “java” or “jvv” or “jaaaa”

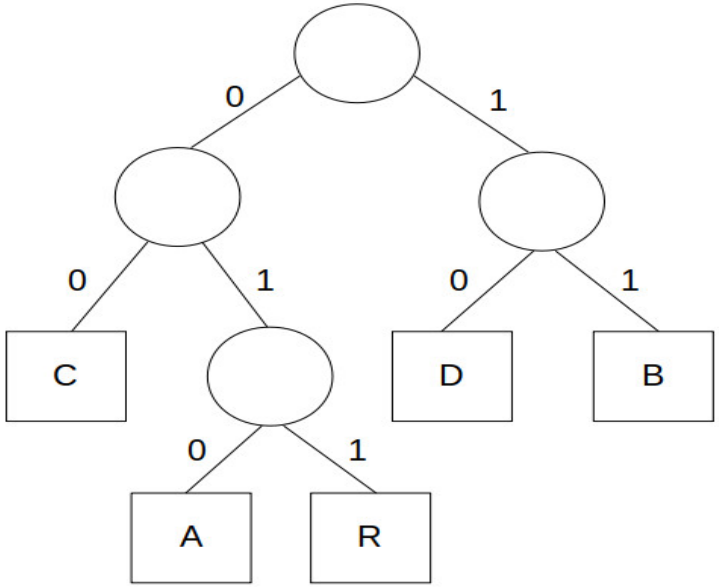
Huffman Encoding Trie

- To prevent ambiguities in decoding, we require that the encoding satisfies the prefix rule: "**No Code is a Prefix of Another**"
- a="0", j="11", v="10" satisfies the prefix rule
- a="0", j="01", v="00" does not satisfy the prefix rule (the code of 'a' is a prefix of the codes 'j' and 'v')

We use an encoding trie to satisfy this prefix rule:

- characters are stored at the external nodes (leaves).
- a left child (edge) means 0.
- a right child (edge) means 1.

Sample Trie



Code: Root to Leaf

A- 010

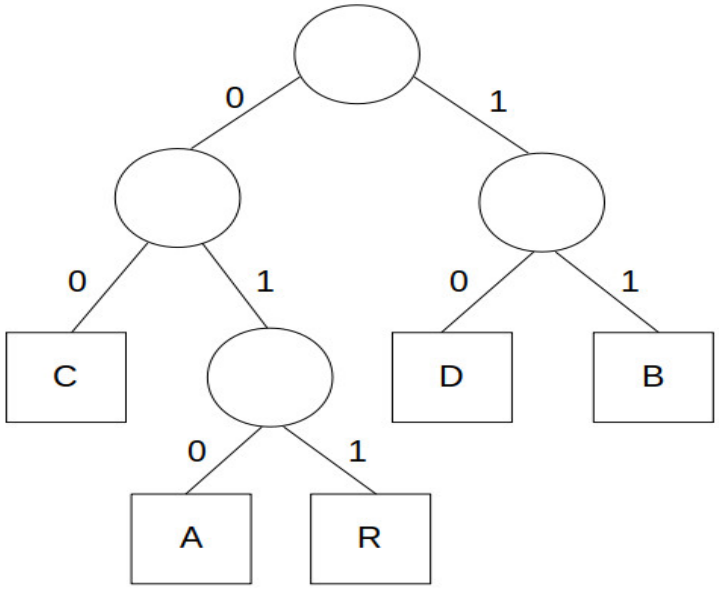
B- 11

C- 00

D- 10

R- 011

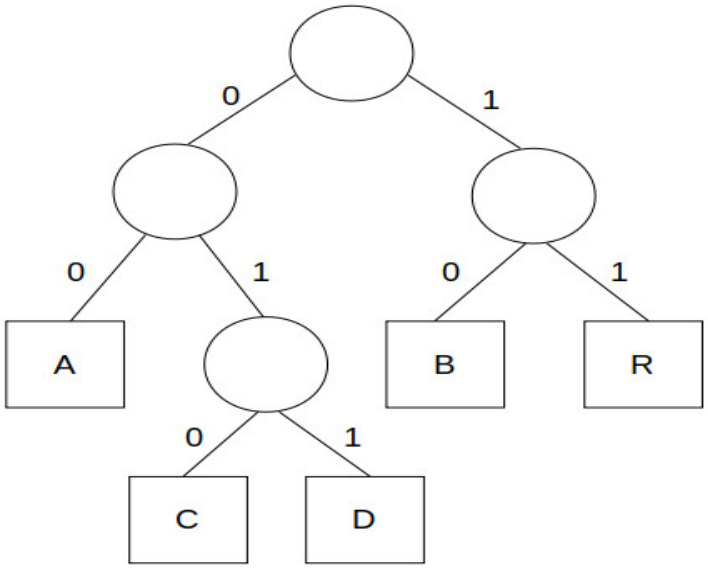
Decoding



A- 010
B- 11
C- 00
D- 10
R- 011
Encoded Text: 01011011010000101001011011010 (29 bits)
Decoded Text: ??

- Previous example: ABRACADABRA: 29 bits
- A- 5 times x 3 bits
- B- 2 x 2
- C- 1 x 2
- D- 1 x 2
- R- 2 x 3

Another trie



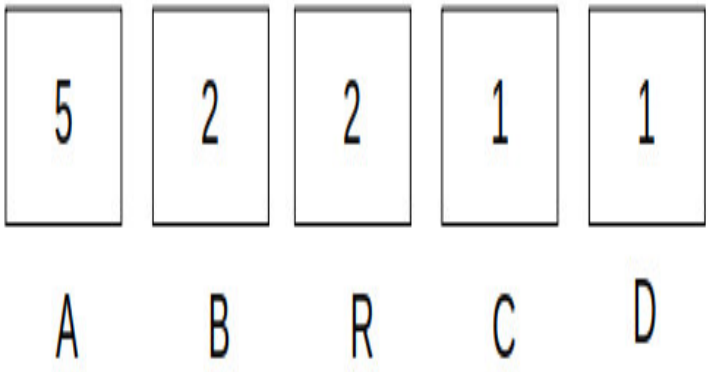
Total Number of bits to encode ABRACADABRA : 24

Optimal Compression

- Ensure that the encoded text is as short as possible.
- Design such a way that the number of bits is as small as possible.
- For this, construct a Huffman encoding trie.

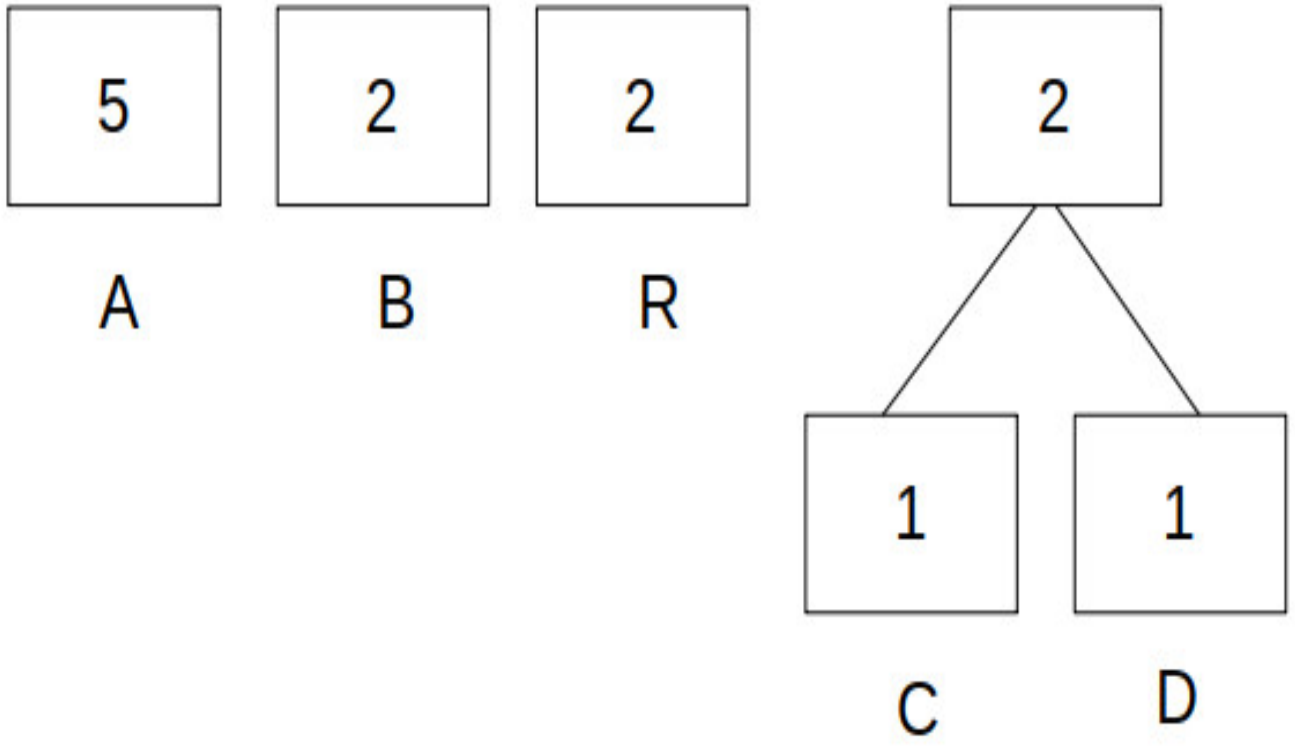
Construction of Huffman Encoding Trie

Sort based on character frequency

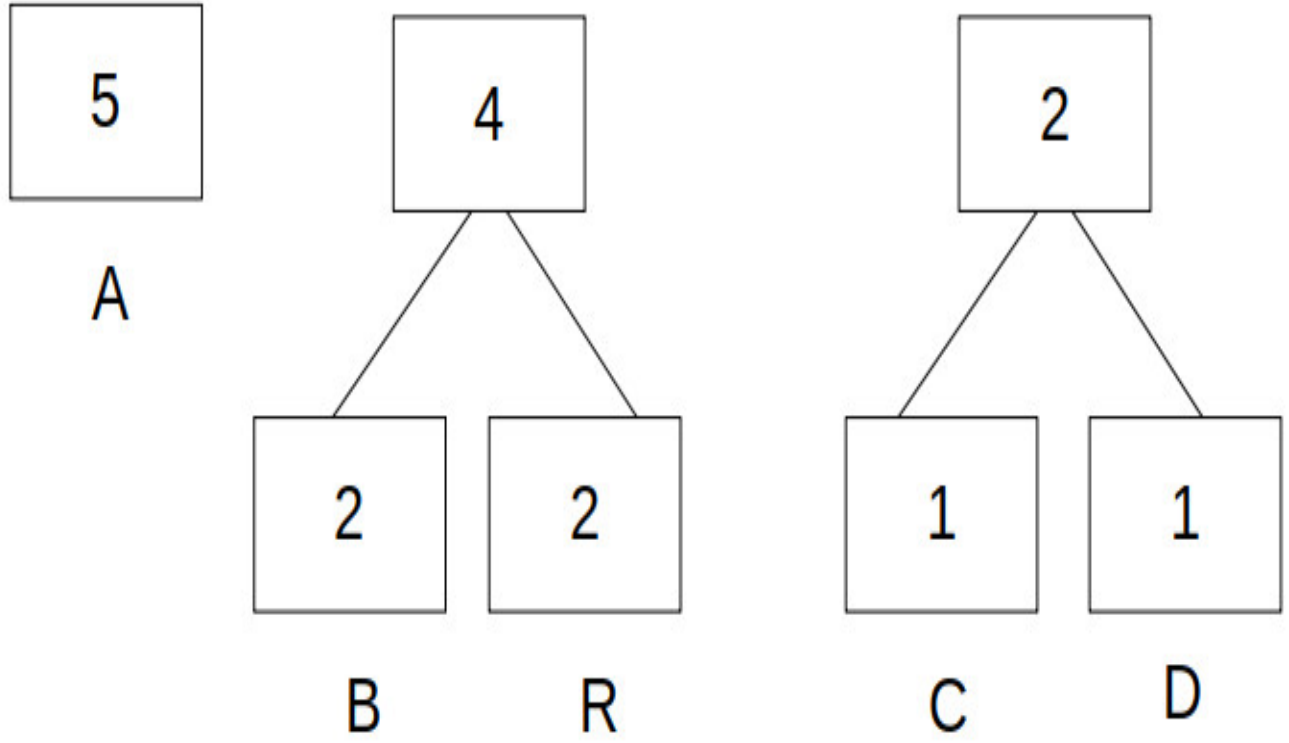


Construction of Huffman Encoding Trie

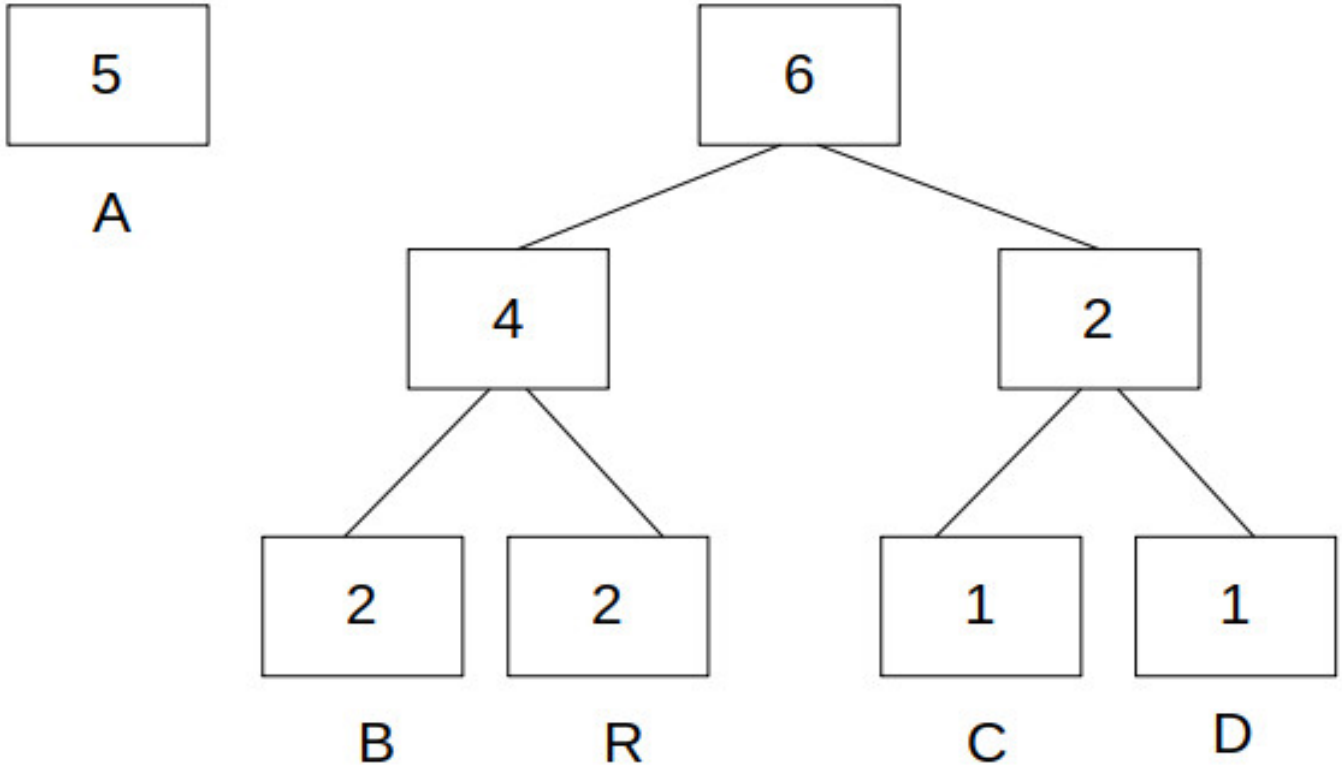
Bring down (one level) the least two: Combine them with their sum as the parent.



Construction of Huffman Encoding Trie

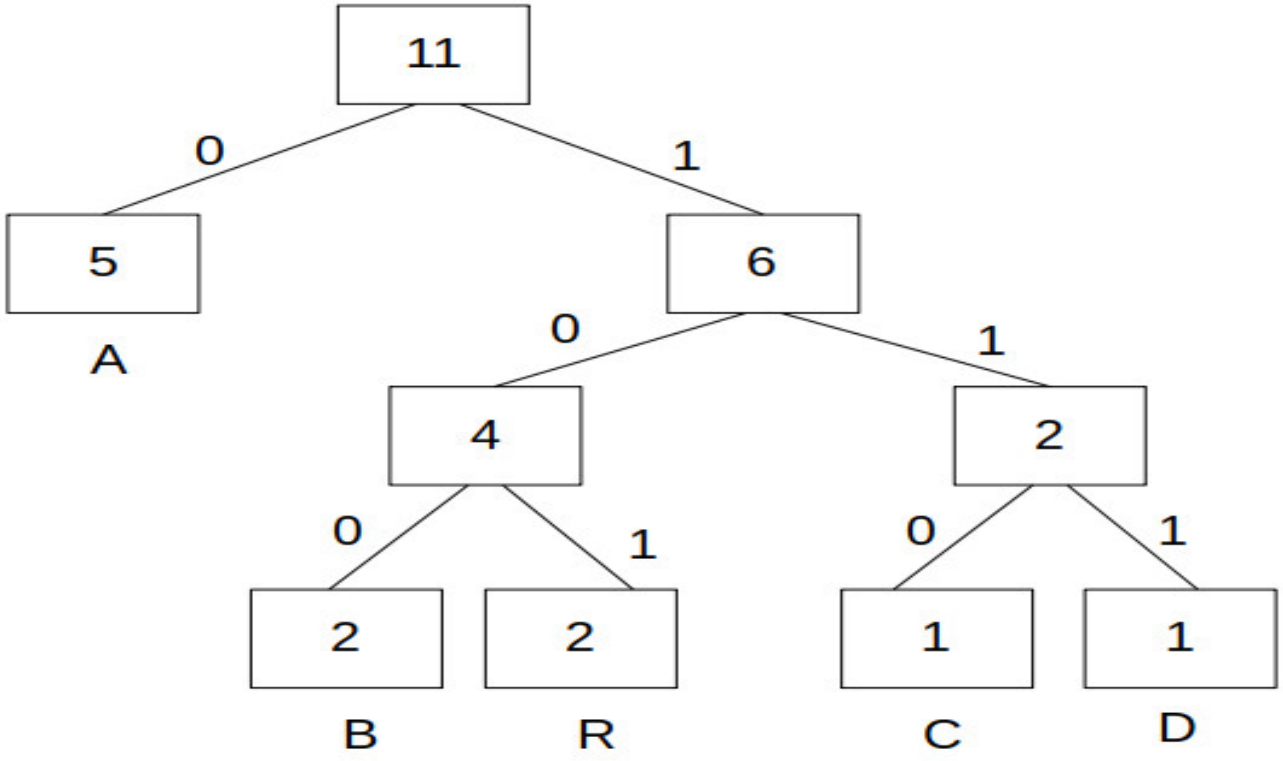


Construction of Huffman Encoding Trie



Construction of Huffman Encoding Trie

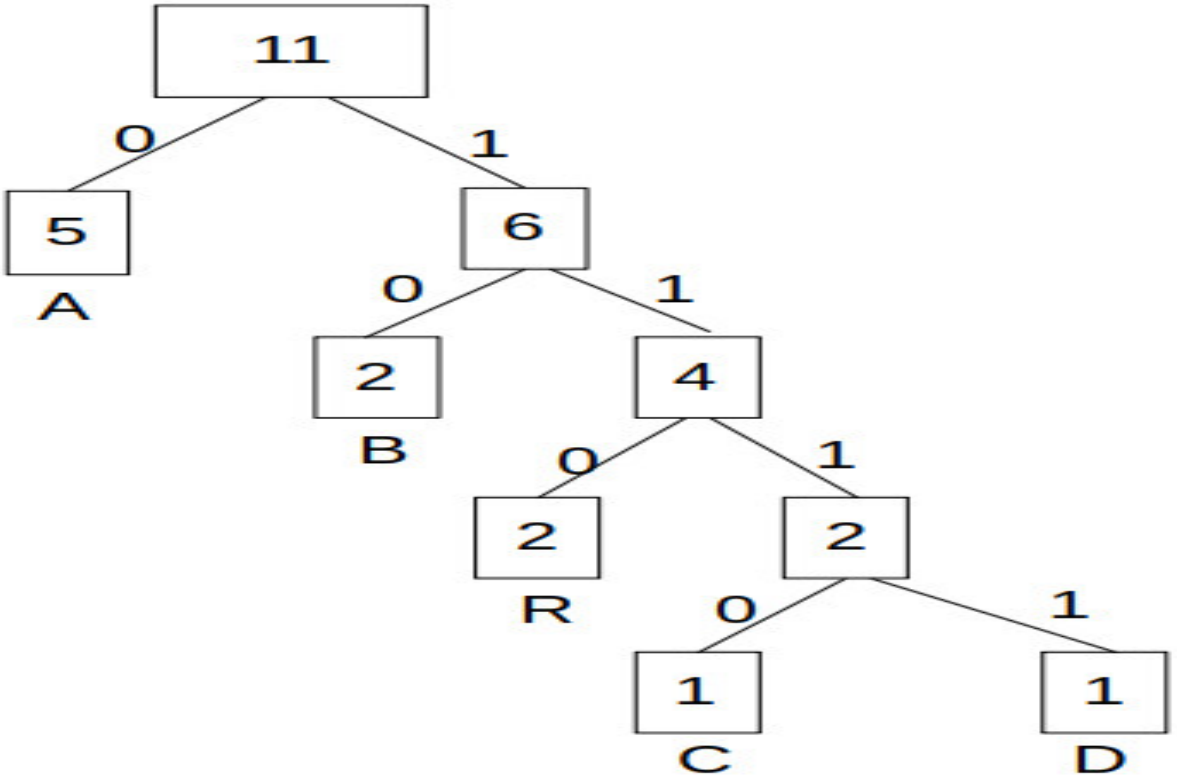
Left edge (Edge to left child): 0
Right edge (Edge to right child): 1



Optimal Compression

- “A” which occurs 5 times gets only one bit.
- ABRACADABRA: 01001010110011101001010 (23 bits) - Best possible

Another Huffman Encoding Trie



ABRACADABRA:01011001100011110101100 (23 bits)

THANK YOU