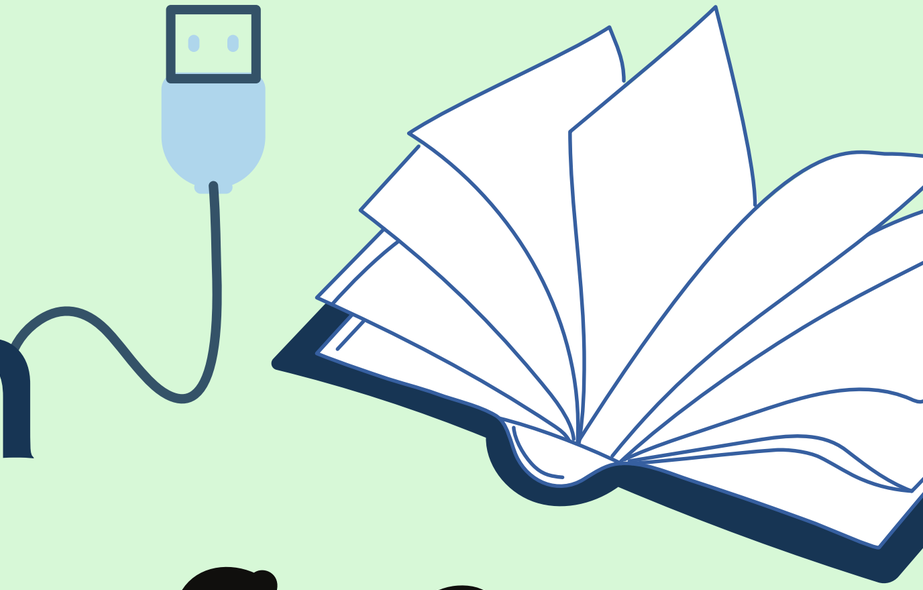
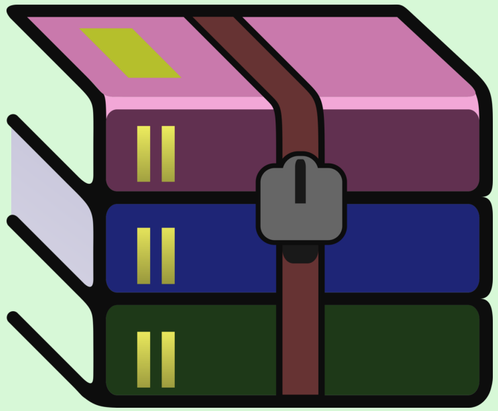
2CS304

Digital Communication



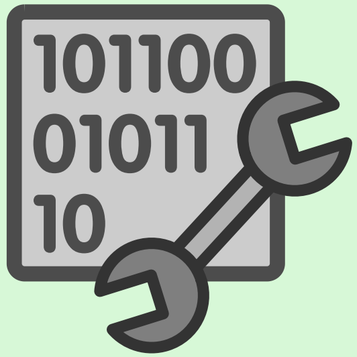
20BCE090 HET PATEL
20BCE099 JAINAM PATEL
20BCE100 JAINIK BAKSHI
20BCE105 JAYNEEL SHAH

LOGO QUIZ



- What does "The Size Solution" mean?

DATA COMPRESSION



Computer stores data in form of 0s and 1s.

Each 0 and 1 would take 1 bit of storage, then how can we compress?

- Each character takes 8 bit (ASCII) .
- Now we cannot change the whole system of representation of characters to compress the text files and we need to cover all the characters as well.
- What we do is...

HUFFMAN CODING

- Huffman Coding is a lossless data compression technique.
- Using Huffman Coding Technique, we would be able to reduce the size of a text file.
- None of the data would be compromised while using this technique.

HUFFMAN CODING

- Computer stores the characters in form of 0s and 1s using 8-bit ASCII Codes.
- for eg: a = 97 = 0110 0001
- if string is apple, for compression we can say that let's say p be 0 and likewise,
- Huffman technique provide us the rules to convert the characters of string to binary digits of less length then the length of its ASCII equivalent.
- The next slide shows the conversion of string "MISSISSIPPI" to Huffman Code, which would occupy less space then the ASCII conversion of the string.

HUFFMAN CODING

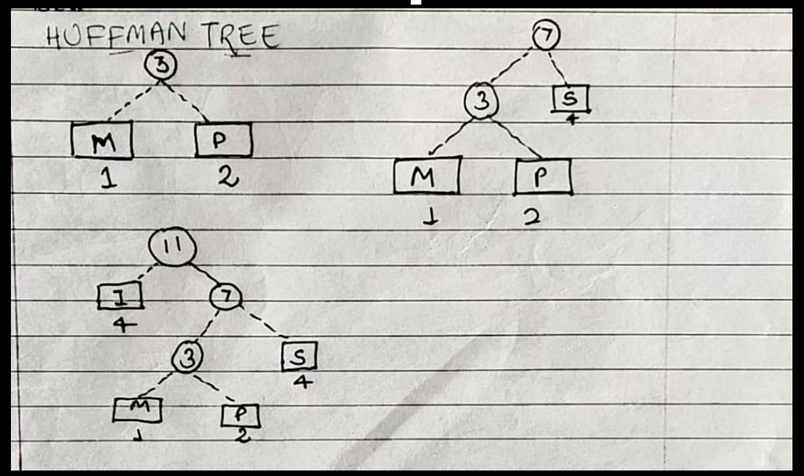
Step 1

MISSISSIPPI

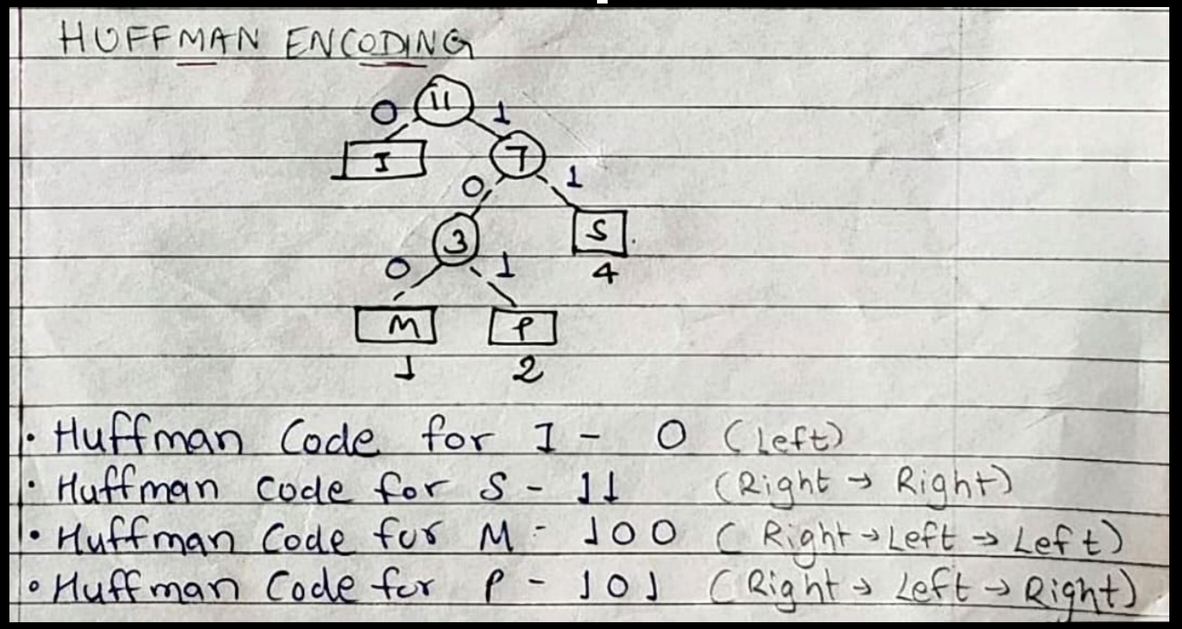
FREQUENCY TABLE

M	1
I	4
S	4
P	2
Total	11

Step 2



Step 3



HUFFMAN CODING

ASCII Encoding

- ASCII encoding (0 - 127)
  - → each character requires 7 bits
  - → For MISSISSIPPI, total characters = 11.
  - → Thus, we rearuire 7×11 = 77 bits.

Random Encoding

- If we know the number of characters, in our case = 4 we can apply a random encoding say - M = 00 I = 01 S = 10 P = 11
  - → In this case every character requires 2 bits to be encoded.
  - → Total bits - = 1×2 + 4×2 + 4×2 + 2×2
  - = 2 + 8 + 8 + 4
  - = 22 bits.

HUFFMAN CODING

Huffman Encoding

- Huffman encoding actually reduces this size too! It assigns more bits to less frequent characters and less bits to highly frequent characters and thus reducing the size further.
→ As we saw above - code for S - 11, code for I = 0, code for M - 100 and code for P = 101.
→ Total bits hence = 1x4 + 2x4 + 1x3 + 2x3
= 4 + 8 + 3 + 6
= 21 bits.

- There is not al significant change in this case as the frequencies are low but it is still lesser than ASCII encoding.

Real Life Applications:

- Text Compression
- ZIP
- Brotli by Google

HUFFMAN CODING

Conclusion

Using, Huffman enoding technique we have successfully changed the bits for the characters of the string, without any loss with respect to the initial string. The size of the file would also decrease and none of the information would be lost as Huffman Coding is lossless.


Thank
you