Huffman Coding

Huffman coding is a popular algorithm used for lossless data compression. It's commonly used in various applications where efficient compression and decompression of data are required.

File Compression and Archiving:

Huffman coding is widely used in file compression algorithms like ZIP, GZIP, and TAR. It allows for efficient compression of files, reducing their size for storage and faster transfer over networks.

Image Compression:

Huffman coding is used in image compression standards like JPEG to reduce the file size of images without significant loss of quality. Huffman coding helps in representing the image data more efficiently.

Video Compression:

Huffman coding is used in video compression standards such as MPEG to reduce the size of video files. It's an essential part of various video codecs, optimizing the encoding and decoding process.

Text Compression:

Huffman coding is used to compress text documents, emails, and other textual data. It assigns shorter codes to more frequent characters or words, optimizing the storage and transmission of text-based information.

Encryption Algorithms:

Huffman coding can be used in encryption algorithms to optimize the representation of encrypted data, making it more efficient for storage or transmission while maintaining security.

Huffman Coding Algorithm

Algorithm: Constructing a Huffman Codes

```
Huffman (C)
1  n ← |C|
2  Q ← C
3  for i ← 1 to n - 1
4       do allocate a new node z
5          left[z] ← x ← Extract-Min (Q)
6          right[z] ← y ← Extract-Min (Q)
7          f [z] ← f [x] + f [y]
8          Insert (Q, z)
9  return Extract-Min(Q)  ♦ Return root of the tree.
```

Character	Frequency
a	5
b	9
c	12
d	13
e	16
f	45

Que – 4. A networking company uses a compression technique to encode the message before transmitting over the network. Suppose the message contains the following characters with their frequency:

Character	Frequency
a	5
b	9
c	12
d	13
e	16
f	45

Note that each character in input message takes 1 byte.

If the compression technique used is Huffman Coding, how many bits will be saved in the message?
(A) 224 (B) 800 (C) 576 (D) 324

Solutions: Finding number of bits without using Huffman,
Total number of characters = sum of frequencies = 100
size of 1 character = 1byte = 8 bits
Total number of bits = 8*100 = 800
Using Huffman Encoding, Total number of bits needed can be calculated as:

5*4 + 9*4 + 12*3 + 13*3 + 16*3 + 45* 1 = 224

Bits saved = 800-224 = 576.

Time complexity:

- O(nlogn) where n is the number of unique characters.
- If there are n nodes, extractMin() is called 2*(n – 1) times.
- ExtractMin() takes O(logn) time as it calls minHeapify().
- So, the overall complexity is O(nlogn).