DATA STRUCTURES & ALGORITHMS

Huffman Algorithm

Instructor: Engr. Laraib Siddiqui

Huffman Algorithm

Every information is encoded as strings of 1s **and** 0s. The objective of information theory is to usually transmit information using fewest number of bits in such a way that every encoding is unambiguous.

Huffman encoding compresses data very effectively saving from 20% to 90% memory, depending on the characteristics of the data being compressed.

Huffman Encoding

- Developed by **David Huffman** in 1951, this technique is the basis for all data compression and encoding schemes
- It is a famous algorithm used for lossless data encoding
- It follows a Greedy approach, since it deals with generating minimum length prefix-free binary codes
- It uses variable-length encoding scheme for assigning binary codes to characters depending on how frequently they occur in the given text. The character that occurs most frequently is assigned the smallest code and the one that occurs least frequently gets the largest code

Example: Huffman’s encoding

A B B C D B C C D A A B B E E E B E A B

Above data contains 20 elements. If we transmit each alphabet using fixed length approach it will take 8 bits each for binary representation and as we have 20 elements, size will be 8 * 20 = 160 bits.

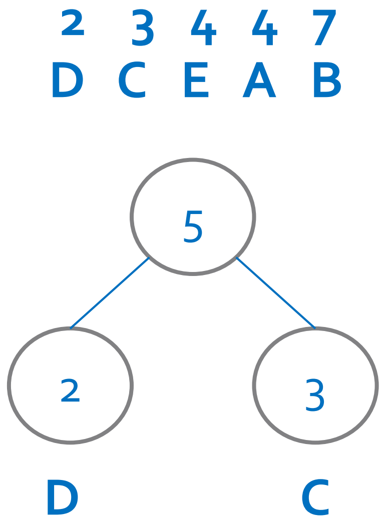
If we use variable length approach/ Huffman encoding then it will be reduced upto 96 bits.

Algorithm

Step 1- Create a leaf node for each character and build a min heap using all the nodes (The frequency value is used to compare two nodes in min heap)
Step 2- Repeat Steps 3 to 5 while heap has more than one node
Step 3- Extract two nodes, say x and y, with minimum frequency from the heap
Step 4- Create a new internal node z with x as its left child and y as its right child. Also frequency(z)= frequency(x)+frequency(y)
Step 5- Add z to min heap
Step 6- Last node in the heap is the root of Huffman tree

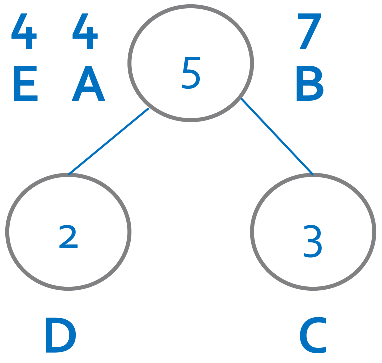
Example: Huffman’s encoding

A B B C D B C C D A A B B E E E B E A B



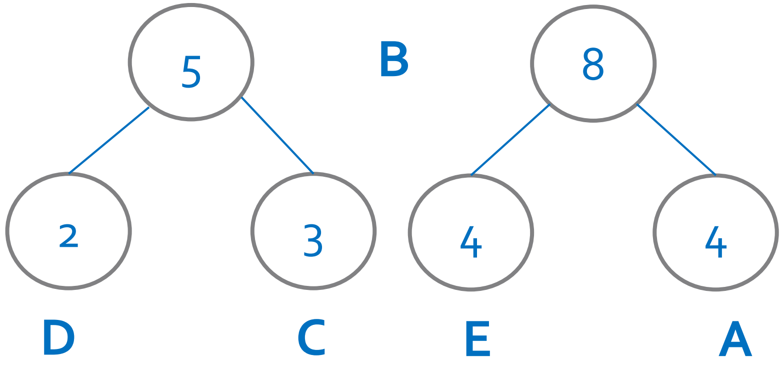
Example: Huffman's encoding

A B B C D B C C D A A B B E E E B E A B



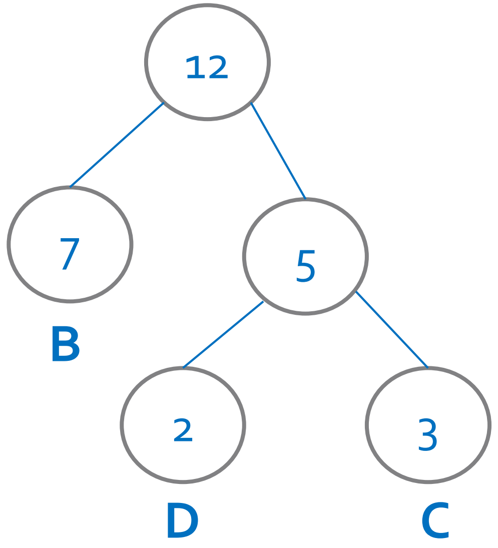
Example: Huffman’s encoding

A B B C D B C C D A A B B E E E B E A B



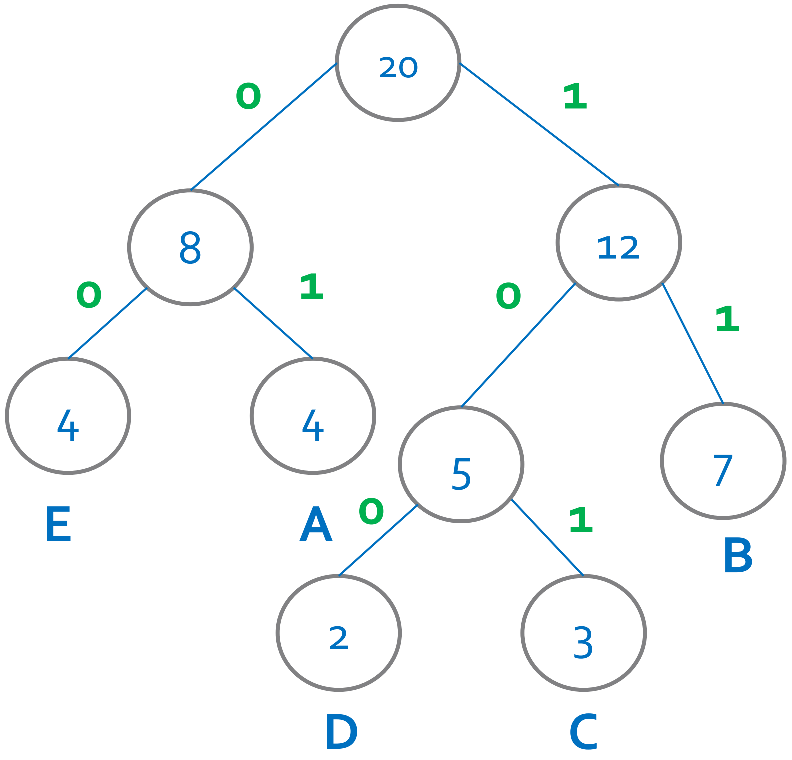
Example: Huffman's encoding

A B B C D B C C D A A B B E E E B E A B



Example: Huffman's encoding

A B B C D B C C D A A B B E E E B E A B



Character	Frequency	Code
A	4	01
B	7	11
C	3	101
D	2	100
E	4	00

Total bits = 97

Practice

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