Huffman coding

submitted by :-
Samyak Bhagat(2018BtechCSE030)
Sanyukta Tanwar(2018BtechCSE026)

Normal Storage pattern

Eg,

message :- BCCABBDDAECCBBAEDDCC

Length of message = 20

ASCII size of each char = 8 bits

Therefore , 8 * 20 = 160 bits to store the data

Huffman Algorithm to compress

- Huffman coding is a lossless data compression algorithm.
- It uses Greedy Approach by using optimal merge pattern
- We use our own defined codes instead of ASCII so as to compress the data

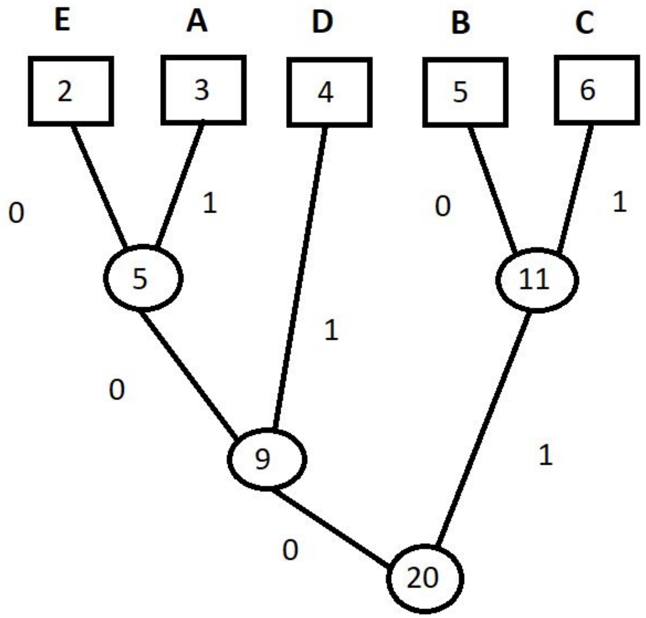
How does this work

Step 1 : Make a frequency count table

Character	**Count**
A	3
B	5
C	6
D	4
E	2

How does this work

Step 2 : Arrange in increasing order of count and make the huffman tree



How does this work

Step 3 : According to tree make the table with new codes

Character	count	code	
A	3	001	3*3 = 9
B	5	10	5*2 = 10
C	6	11	6*2 = 12
D	4	01	4*2 = 8
E	2	000	2*3 = 6

Therefore , encoded message will take 45 bits and the memory for the table

Time complexity

Huffman coding use a heap to store the weight of each tree, each iteration requires **O(logn)** time to determine the cheapest weight and insert the new weight.

There are **O(n)** iterations, one for each item. Therefor the time complexity of the Huffman algorithm is **O(nlogn)**.

Actual File Compression Size

When we tested the actual file size for sample text was **716.7 KB**

And when we compressed it the file size was reduced to **394 KB** which included both the huffman code table along with encoded text

Therefore in this case the file was compressed to abot **54%**