

Assignment 2

Rajdhani

DATE / /

Title:- Huffman Encoding using greedy strategy.

Aim:-

Write a program to implement Huffman Encoding using a greedy strategy.

Requirements:-

- 64 bit OS
- Ubuntu / Linux
- 8 GB RAM ; ;

Objectives:-

- Understand and implement the concept of Huffman encoding.

Theory:-

Huffman Encoding:

Huffman encoding is a lossless data compression algorithm.

The idea is to assign ~~variable-length~~ codes to input characters. The length assigned codes are based on the frequency of corresponding characters.

The most frequent character gets the smallest code and the least frequent character gets the smallest code.

There are mainly two major parts in Huffman encoding:-

- 1) Build a huffman tree from input characters.
- 2) Traverse the Huffman tree & assign codes to characters.
- 3) Steps to build Huffman tree-

Input is an array of unique characters along with their frequency of occurrence and output is Huffman tree.

- 4) Create a leaf node for each unique characters and build a min heap is used as a priority queue.
The value of frequency field is used to compare two nodes in a min heap, initially the least frequency character is at root.
- 5) Extract two nodes with the minimum frequency from the min heap.
- 6) Create a new internal node with a frequency equal to the sum of the nodes frequency.
Make the first extracted node as its right child and the other extracted node as its left child. Add this node to min heap.
- 7) Repeat step 2 & 3 until the heap contains only one node. The remaining node is the root node and the tree is complete.

Huffman Encoding algorithm:-

Step 1: Create a priority queue Q consisting of each unique character.

Step 2: Send them in ascending order of their

Frequencies.

Step 3: For all the unique characters -

- 1) Create a new node.
- 2) Extract min value from Q & assign it to left child of new node.
- 3) Extract min value from Q & assign it to right child of new node.
- 4) Calculate the sum of these two minimum values this new node into the trace return new node.
- 5) Insert this node into the trace return root node.

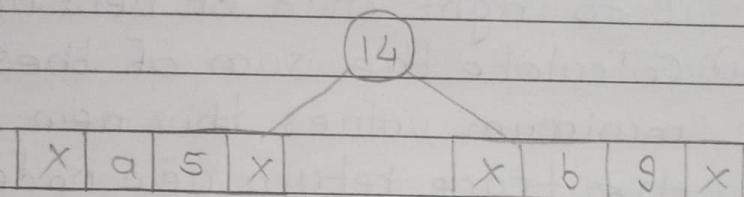
Analysis:-

The time complexity for encoding each unique character based on its frequency is $O(n \log n)$. Extracting minimum frequency from the priority queue take place $2 * (n-1)$ times & its complexity is $O(n \log n)$. Thus, the overall complexity is $O(n \log n)$.

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

Step 1: Build a min heap that contains 6 nodes where each node represents root of a tree with single node

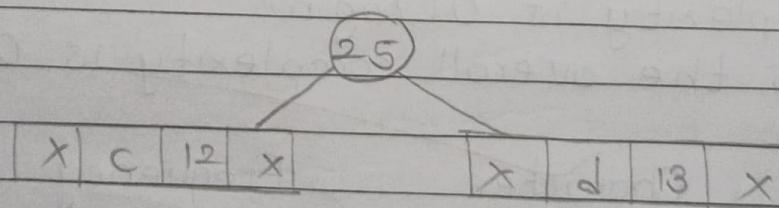
Step 2: Extract two minimum frequency from min heap. Add a new internal node with frequency $5+9=14$.



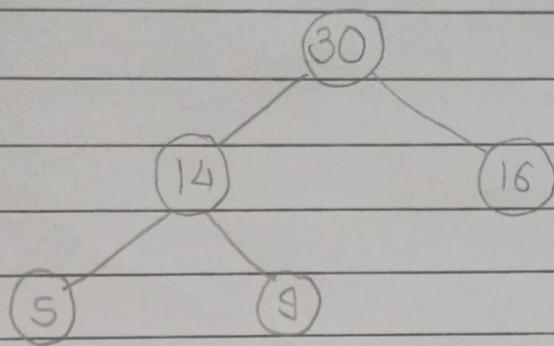
Now, min heap contains 5 nodes where 4 nodes are root of trees with single element each and one heap node is root of tree with 3 elements.

Step 3: Extract two minimum frequency nodes from heap.

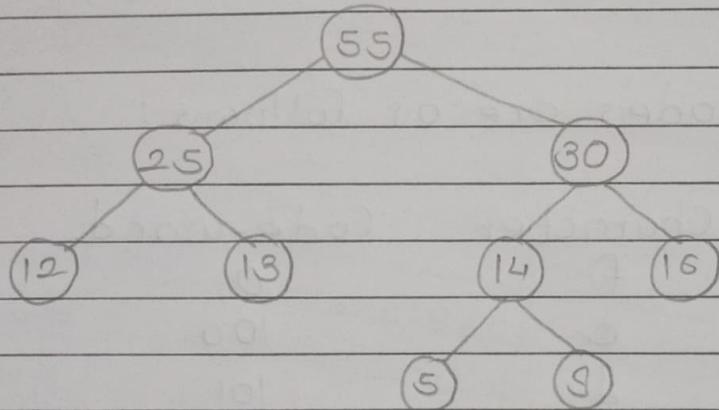
Add a new internal node with frequency $12+13=25$.



Step 4: Extract two minimum frequency nodes. Add a new internal node with frequency $14+16=30$.

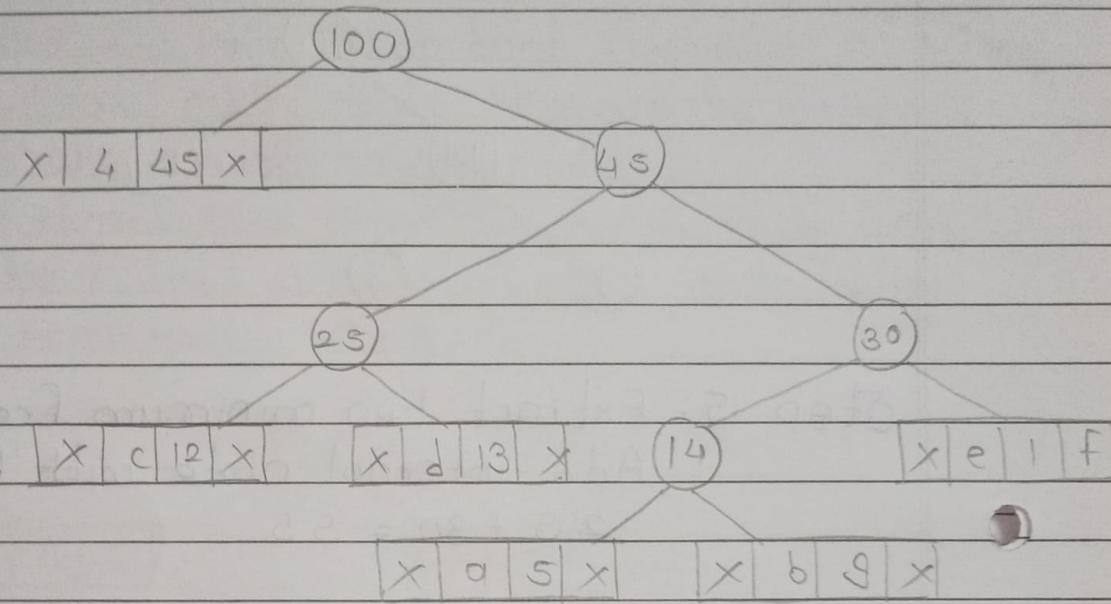


Step 5: Extract two minimum frequency nodes.
Add internal node with frequency
 $25 + 30 = 55$



Step 6: Extract two minimum frequency nodes.
Add new internal node with
frequency $45 + 55 = 100$.

Steps to print codes from Huffman tree:-
Traverse the tree formed string from the root. in auxiliary array while moving to the left child, 0 to then array



The codes are as follows :

Character	Code-word.
f	0
c	100
d	101
a	1100
b	1101
e	111

Conclusion :-

Hence, we understood how to implement Huffman coding using a greedy strategy with example.

~~No fear~~