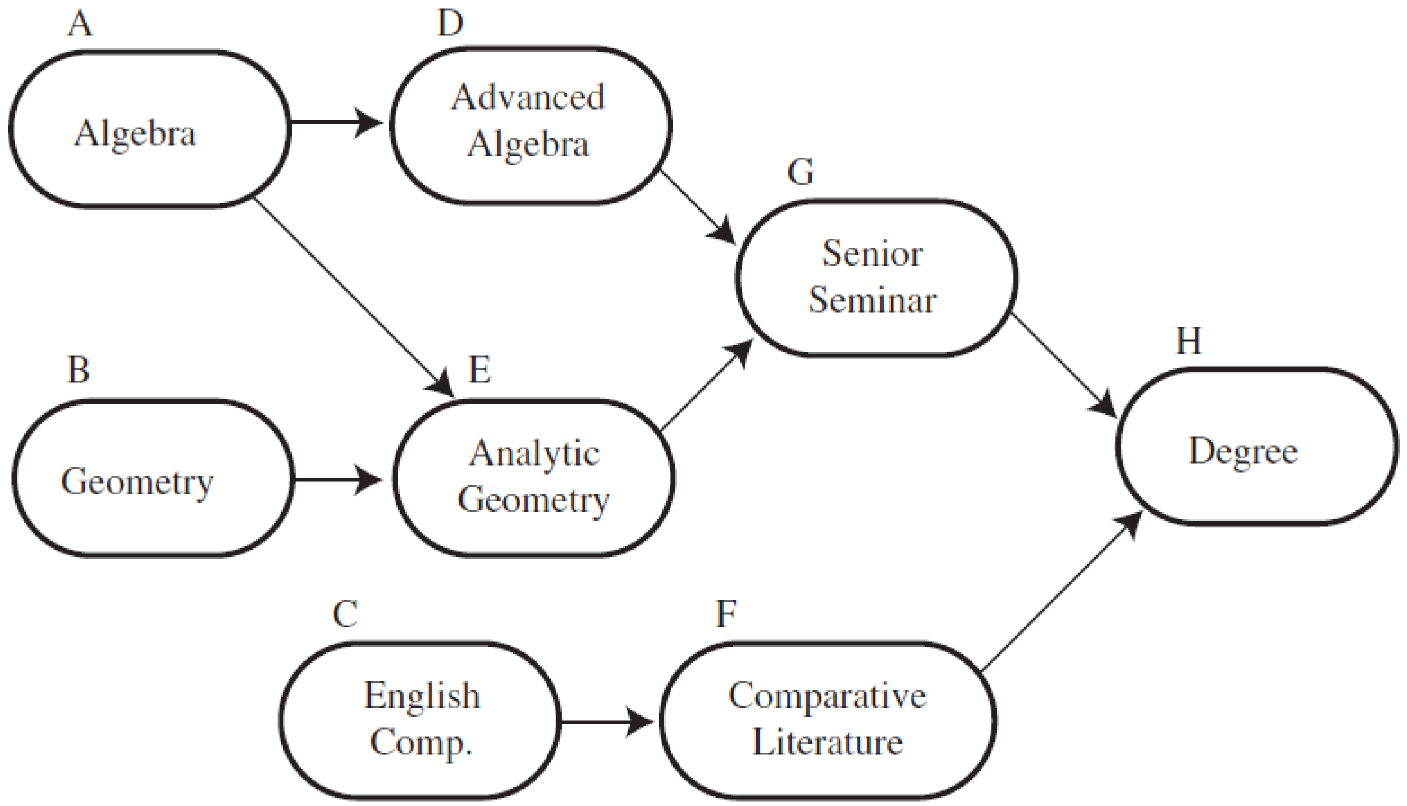
TOPICS

Topological Sorting

Huffman Coding

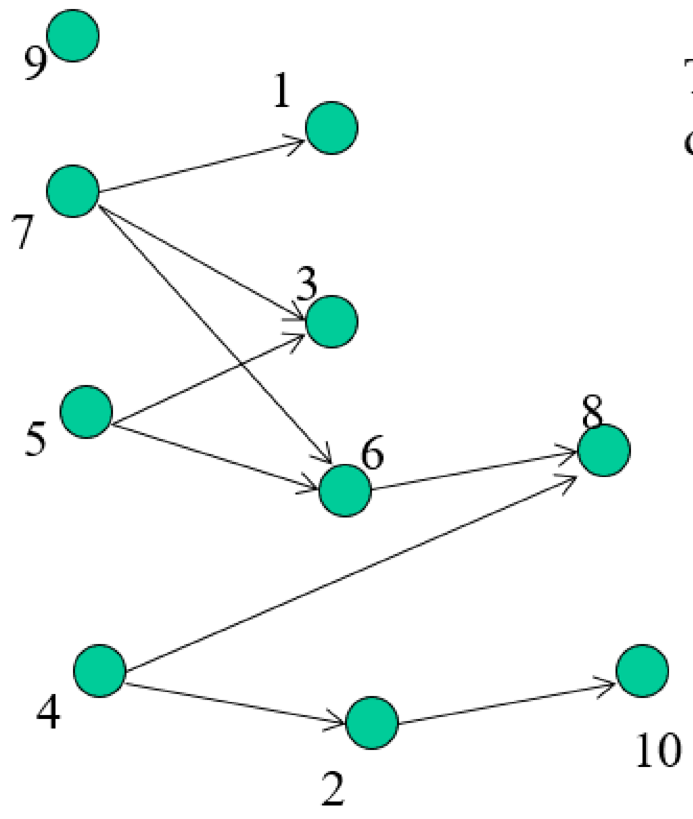
Topological Sorting with Directed Graphs

Topological sorting is another operation that can be modeled with graphs. It’s useful in situations in which items or events must be arranged in a specific order.



Topological Sort Example

- A job consists of 10 tasks with the following precedence rules:
- Must start with 7, 5, 4 or 9.
- Task 1 must follow 7.
- Tasks 3 & 6 must follow both 7 & 5.
- 8 must follow 6 & 4.
- 2 must follow 4.
- 10 must follow 2.



Tasks shown as a directed graph.

First Year - First Semester

Course Name	Descriptive Title	Credit Unit			No. of Hrs/Week			Pre-Requisite	Co-Requisite
		Lec	Lab	Total	Lec	Lab	Total		
ICT 102	INTRODUCTION TO COMPUTING	2	1	3	2	3	5		
ICT 103	FUNDAMENTALS OF PROGRAMMING WITH PROGRAM LOGIC FORMULATION	2	1	3	2	3	5		
GE 1 SS	UNDERSTANDING THE SELF	3	0	3	3	0	3		
GE 4 MATH	MATHEMATICS IN THE MODERN WORLD	3	0	3	3	0	3		
GE 5 ENG	PURPOSIVE COMMUNICATION	3	0	3	3	0	3		
GE 8 SS	ETHICS	3	0	3	3	0	3		
PE 1	EDUCATIONAL GYMNASTICS	2	0	2	2	0	2		
NSTP 1	ROTC 1/LTS 1/CWTS 1	3	0	3	3	0	3		
	Total	**21**	**2**	**23**	**21**	**6**	**27**		

First Year - Second Semester

Course Name	Descriptive Title	Credit Unit			No. of Hrs/Week			Pre-Requisite	Co-Requisite
		Lec	Lab	Total	Lec	Lab	Total		
ICT 104	INTERMEDIATE PROGRAMMING	2	1	3	2	3	5	ICT 103	
~~ICT 105~~	~~DISCRETE STRUCTURE 1~~	3	0	3	3	0	3	CS 1	
ICT 106	SYSTEM FUNDAMENTALS	3	0	3	3	0	3	ICT 102	
GE 2 SS	READINGS IN PHILIPPINE HISTORY	3	0	3	3	0	3		
ENG 3	TECHNICAL WRITING WITH ORAL COMMUNICATION	2	1	3	2	3	5		
GE 3 SS	THE CONTEMPORARY WORLD	3	0	3	3	0	3		
GE ELEC 9	PHILIPPINE POPULAR CULTURE	3	0	3	3	0	3		
PE 2	INDIVIDUAL/DUAL SPORTS	2	0	2	2	0	2	PE 1	
NSTP 2	ROTC 2/LTS 2/CWTS 2	3	0	3	3	0	3	NSTP 1	
	Total	**24**	**2**	**26**	**24**	**6**	**30**		

Objectives

What are Huffman Codes ?

Why Huffman Codes are used ?

Introduction

An effective and widely used Application of *Binary Trees* and *Priority Queues*

Developed by *David.A.Huffman* while he was a Ph.d student at MIT and published in the *1952* paper *"A Method for the Construction of Minimum Redundancy Codes"*.

Each code is a *binary string* that is used for transmission of the corresponding message.

B	1
A	2
G	3
E	1

For Example :

BAGGAGE $\longrightarrow$ 100 11 0 0 11 0 101

PRACTICE PROBLEM BASED ON HUFFMAN CODING-

Problem-

A file contains the following characters with the frequencies as shown. If Huffman Coding is used for data compression, determine-

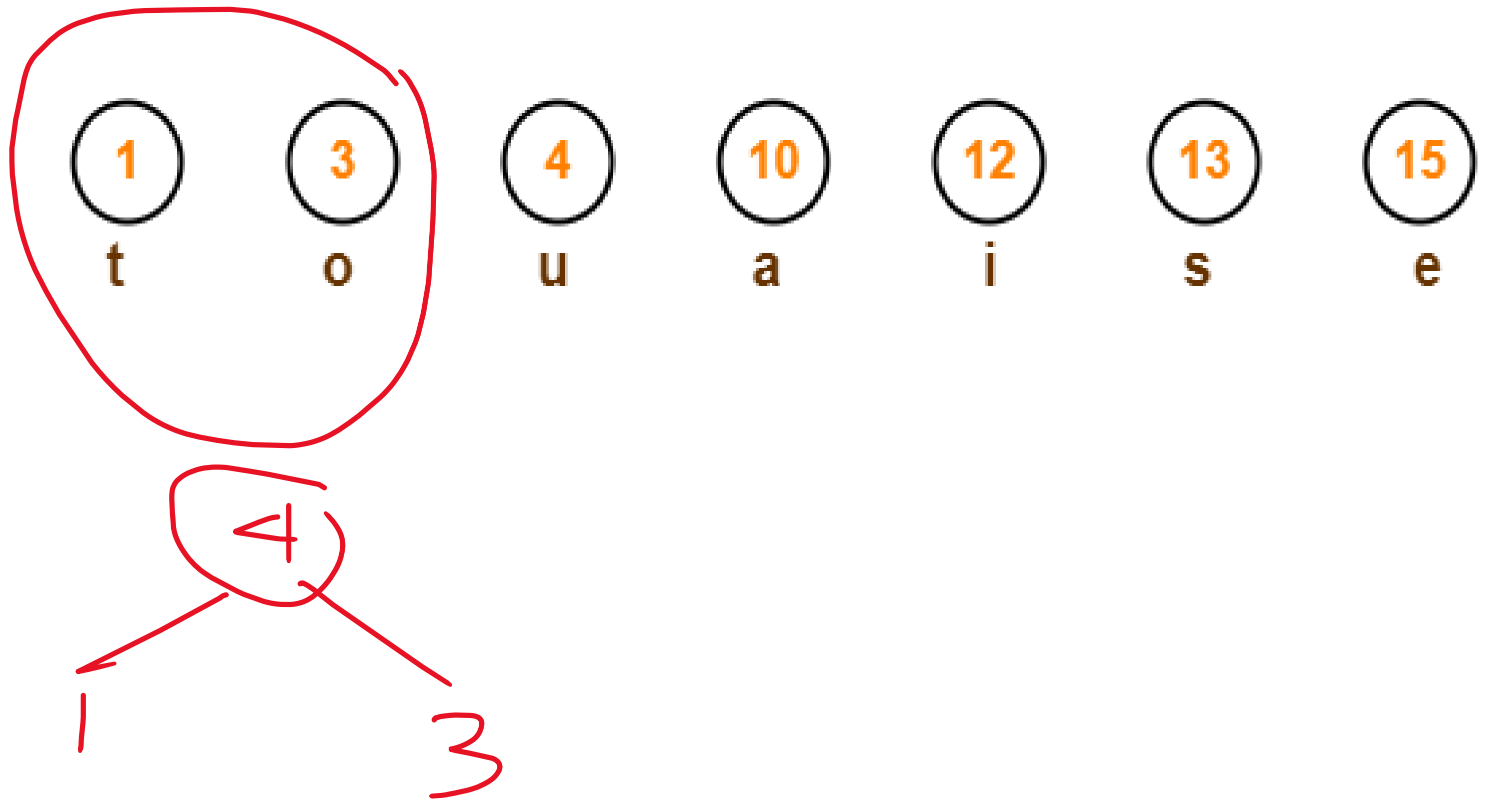
1. Huffman Code for each character
2. Average code length
3. Length of Huffman encoded message (in bits)

Characters	Frequencies
a	10
e	15
i	12
o	3
u	4
s	13
t	1

Solution-

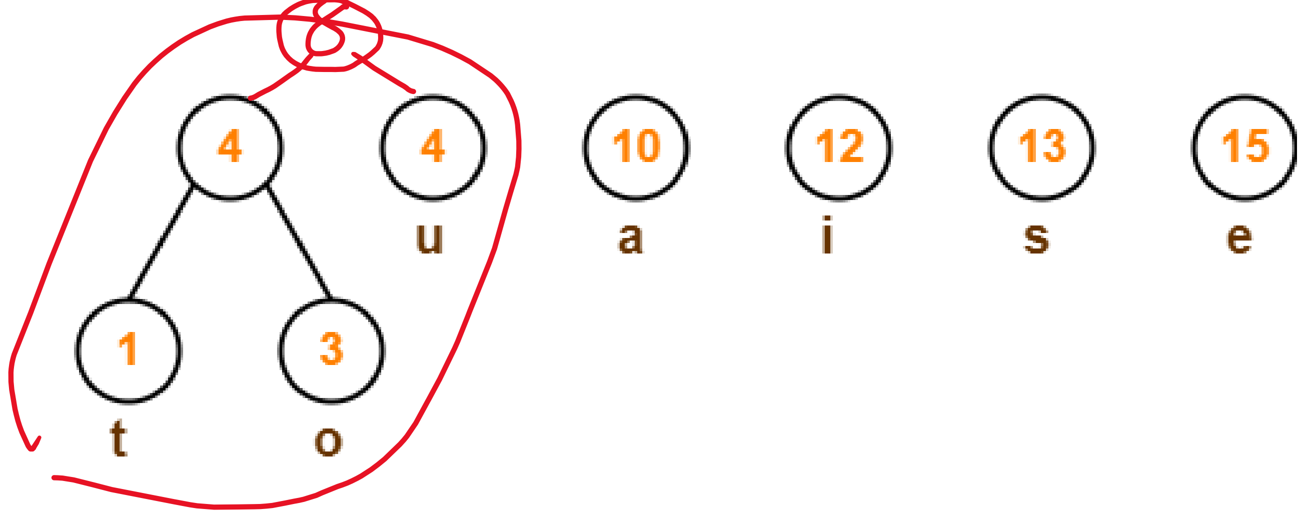
First let us construct the Huffman Tree.
Huffman Tree is constructed in the following steps-

Step 1:

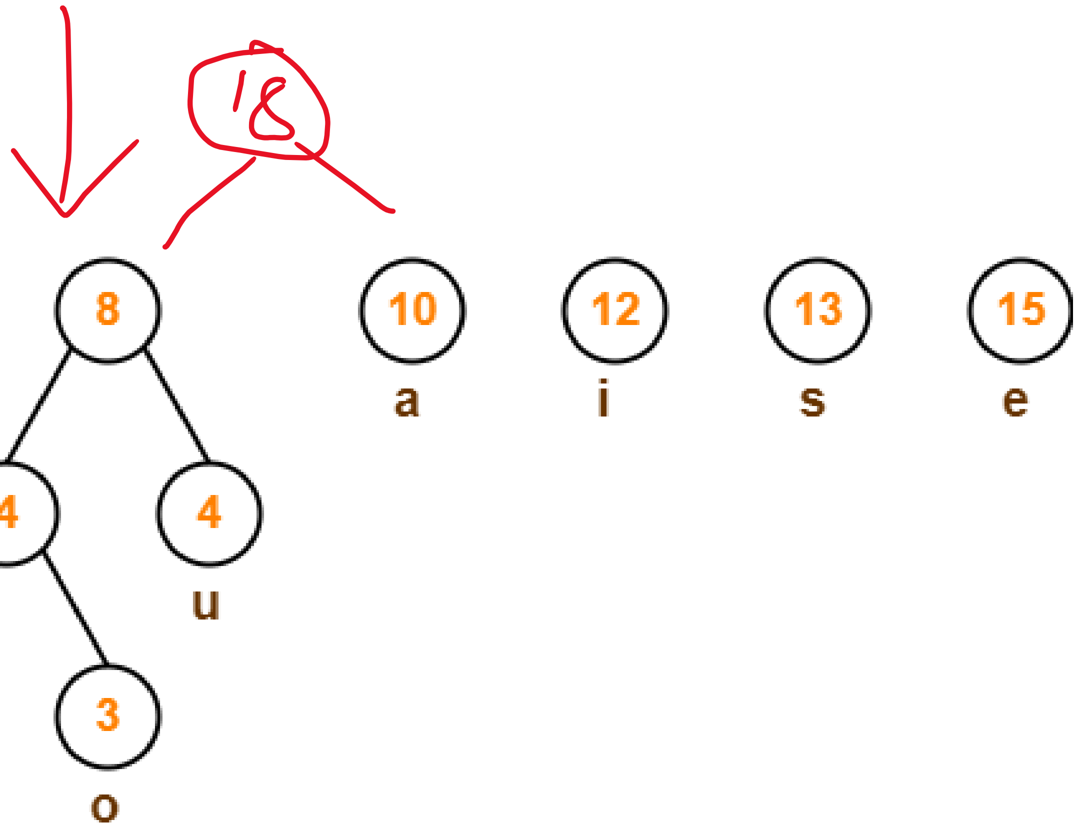


Characters	Frequencies
a	10
e	15
i	12
o	3
u	4
s	13
t	1

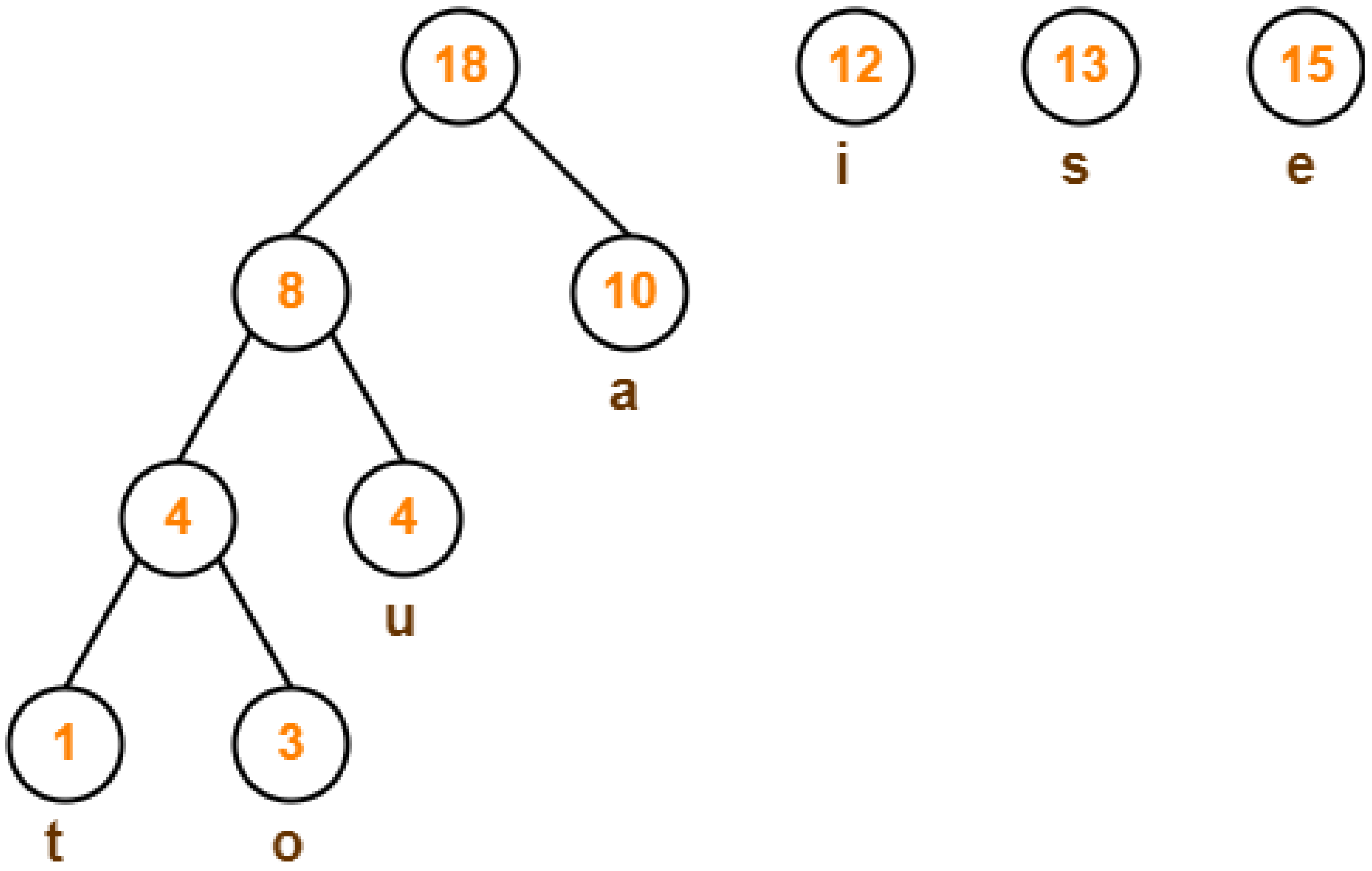
STEP 2



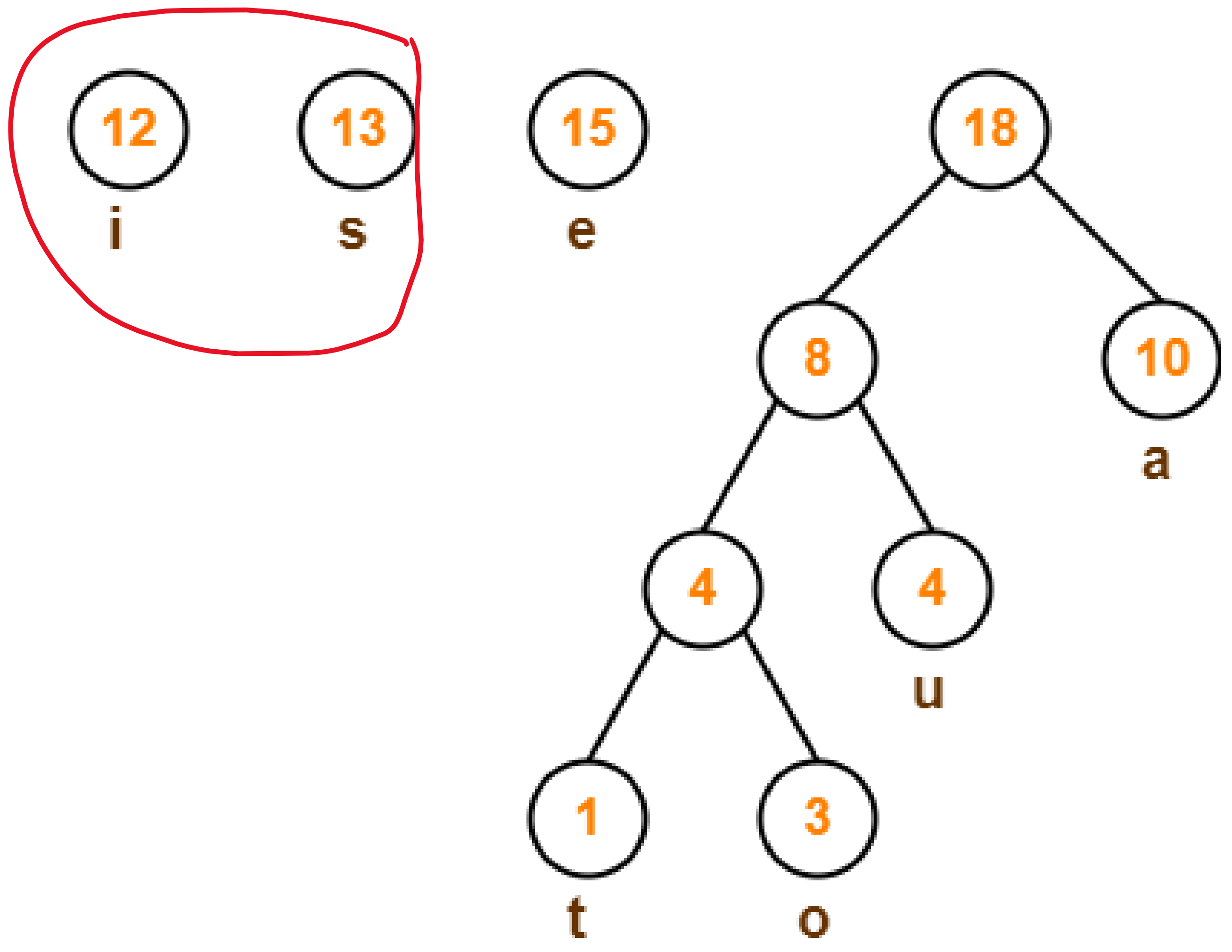
STEP 3



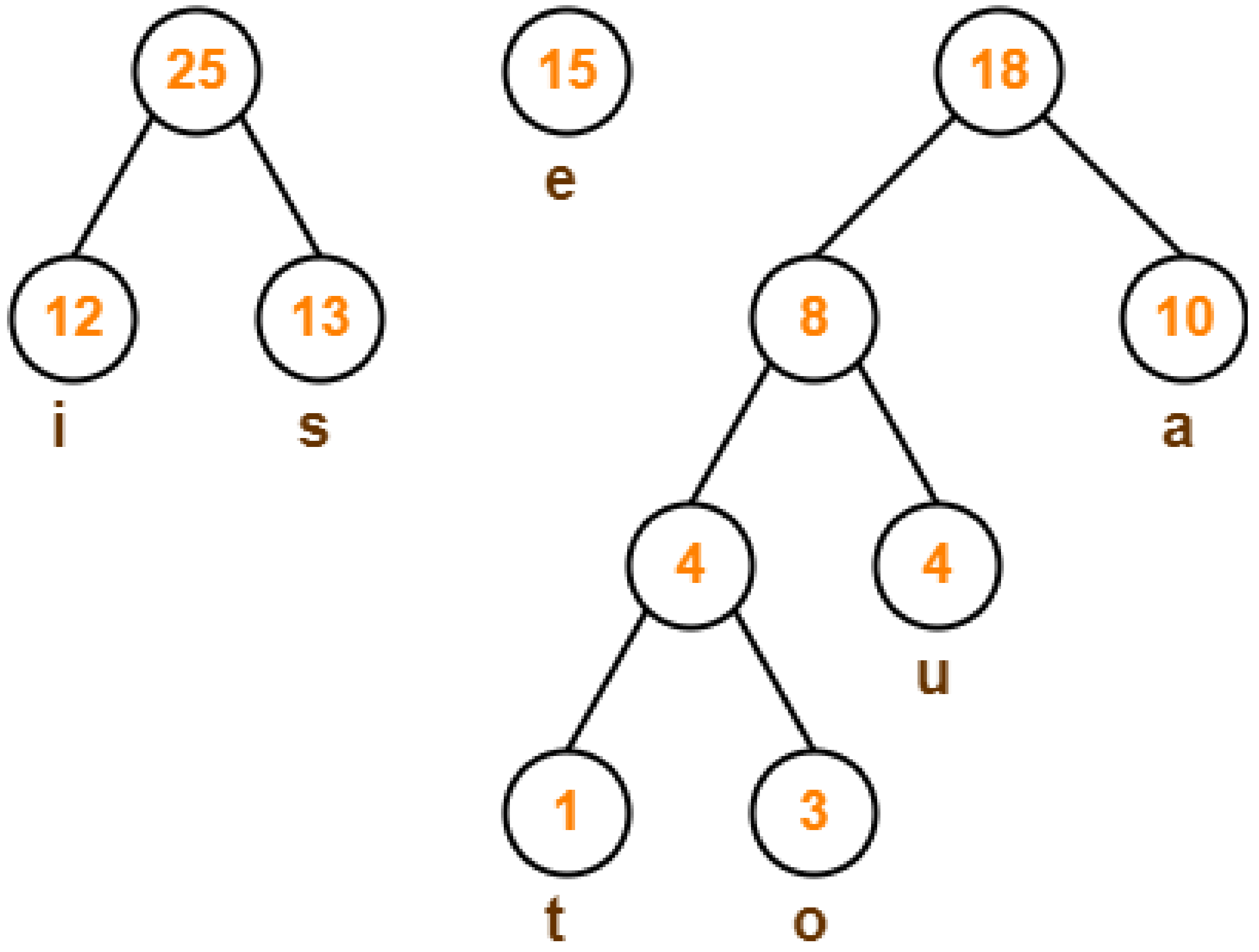
STEP 4



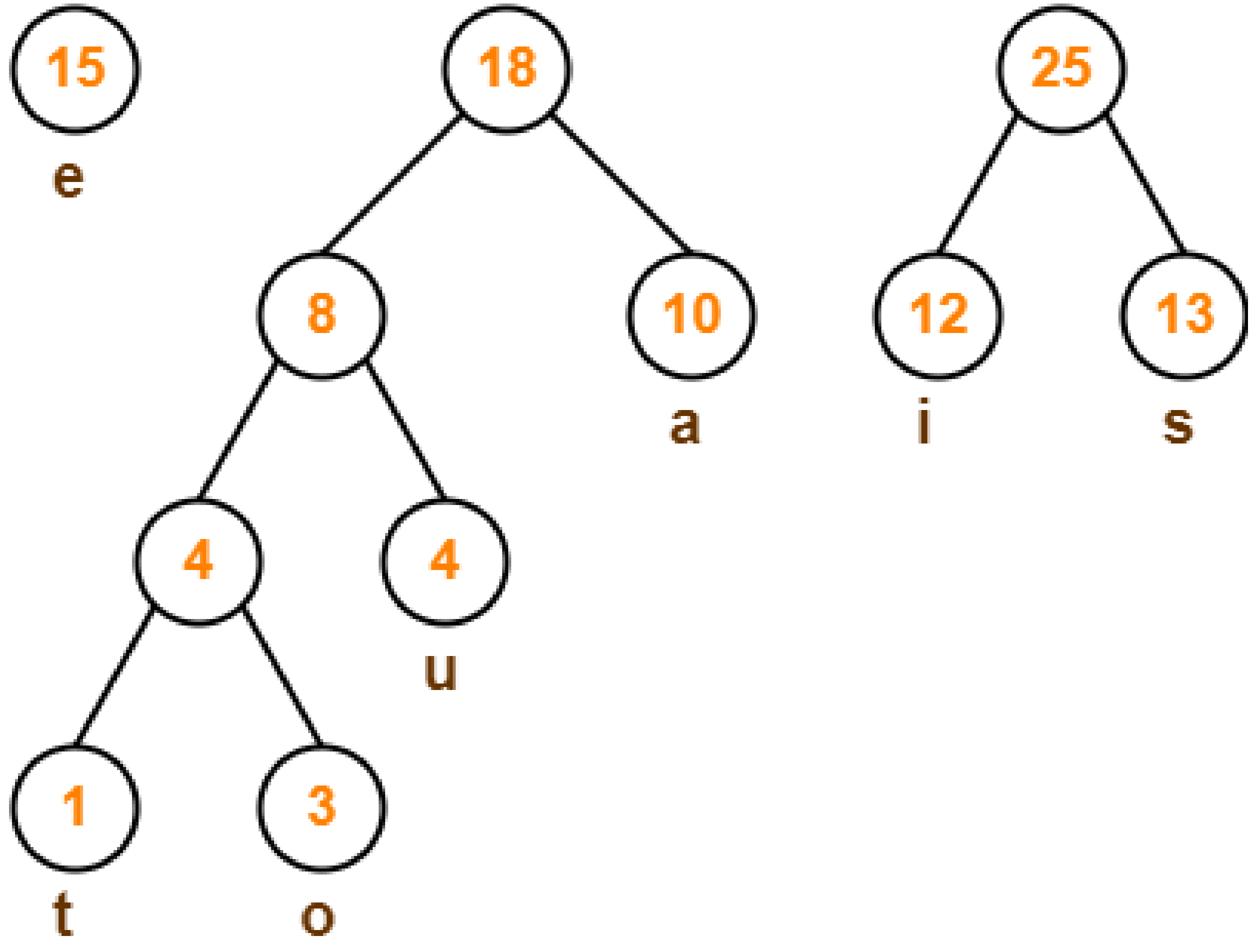
STEP 5



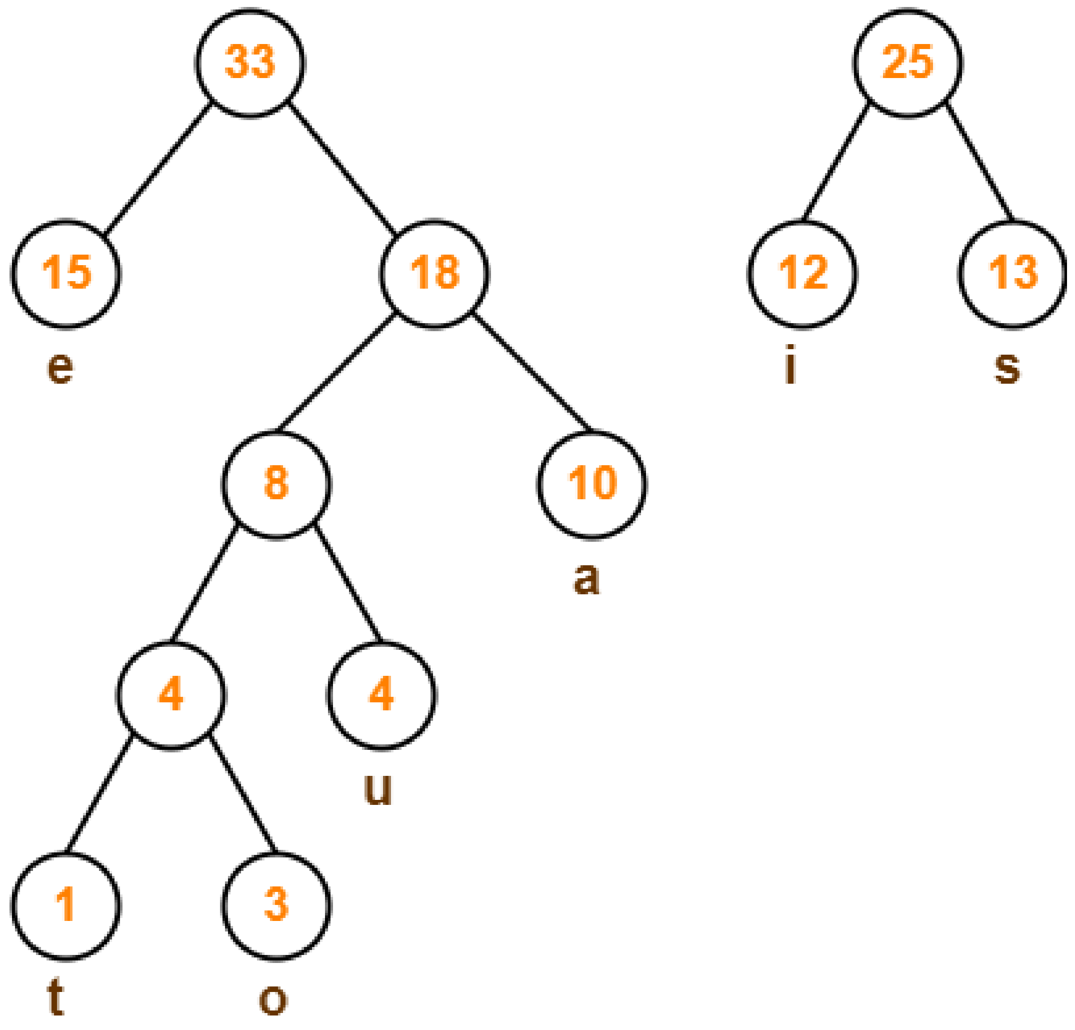
STEP 6



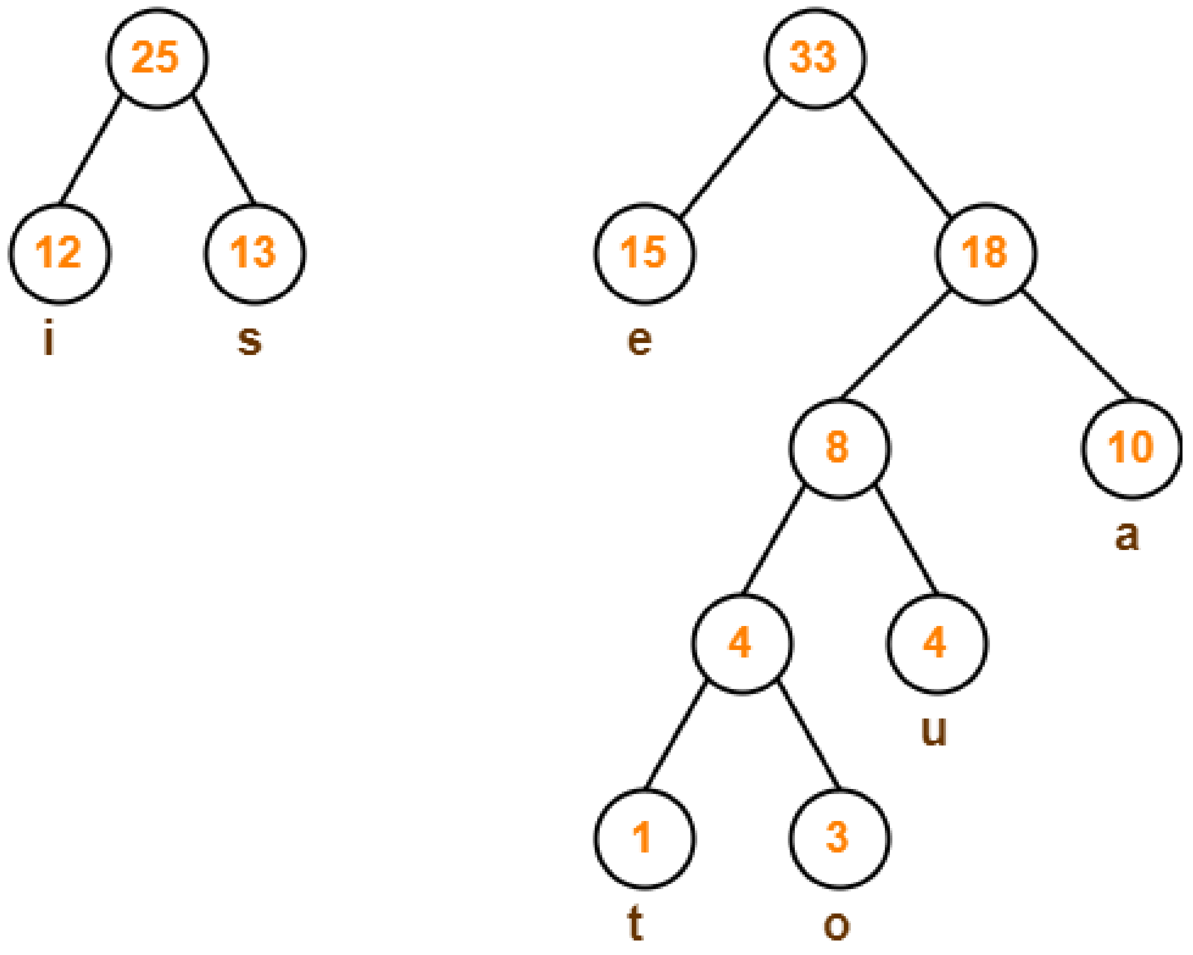
STEP 7



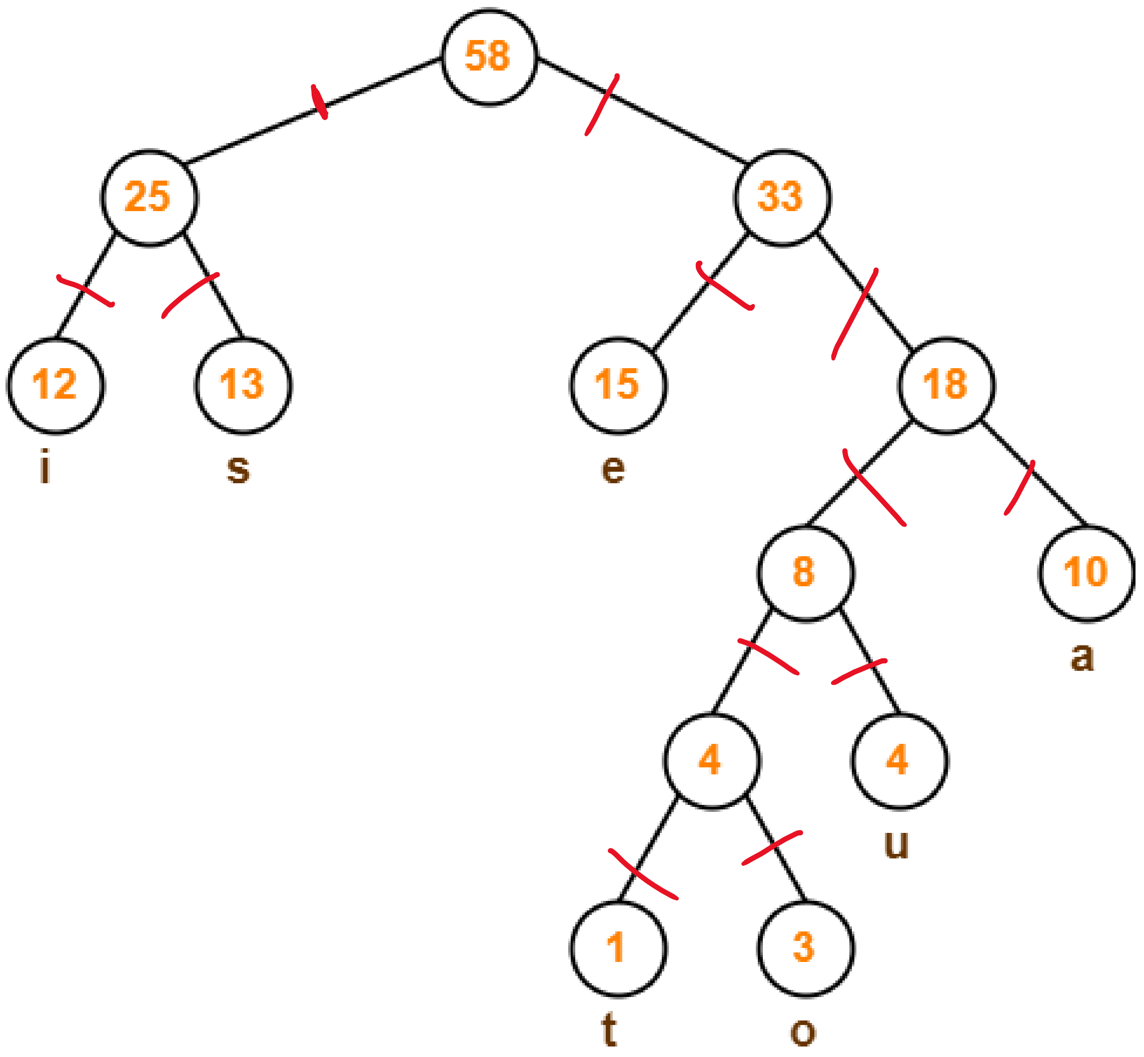
STEP 8



STEP 9



STEP 10

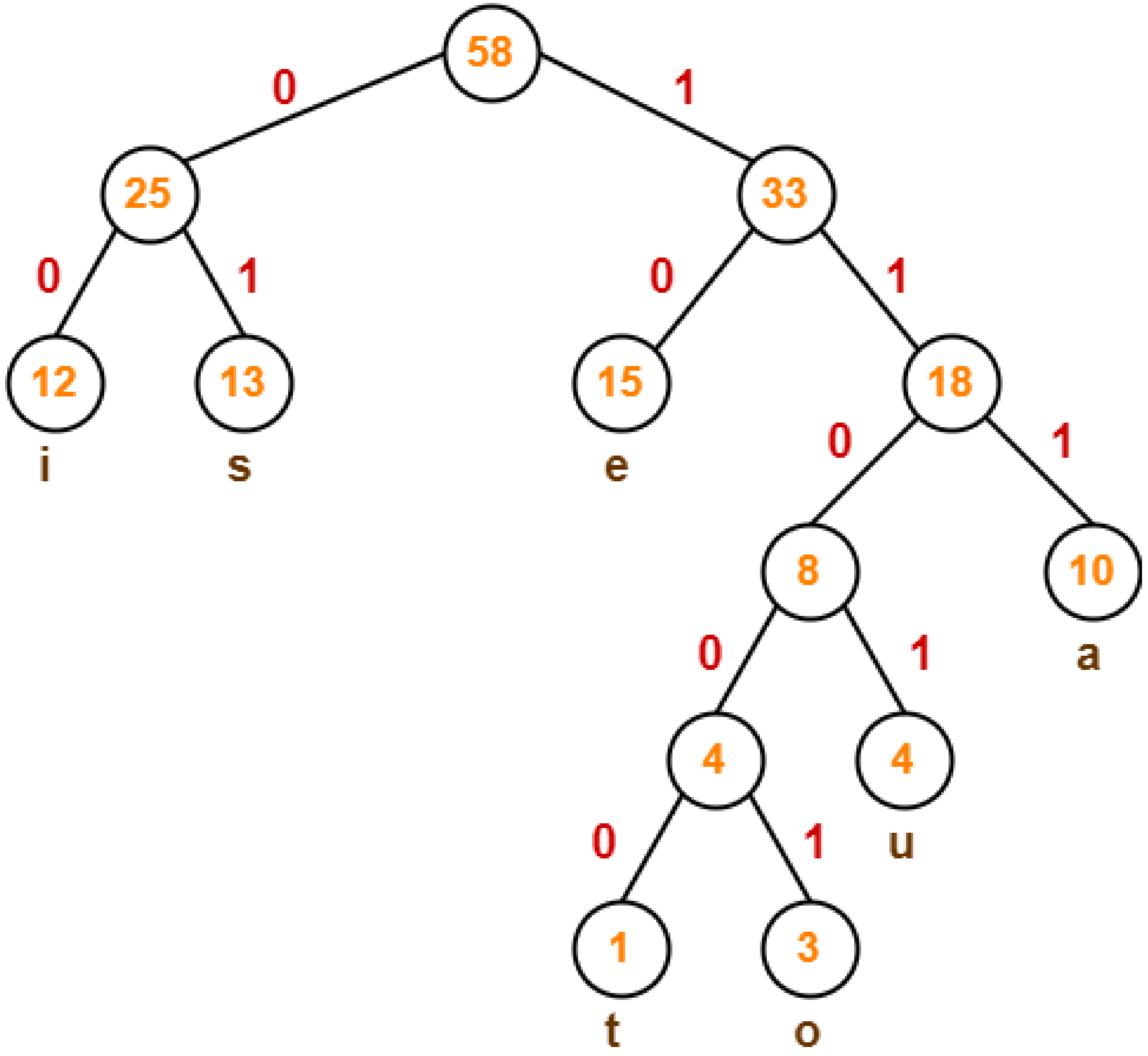


Huffman Tree

- We assign weight to all the edges of the constructed Huffman Tree.
- Let us assign weight ‘0’ to the left edges and weight ‘1’ to the right edges.

<u>Rule</u>

- If you assign weight ‘0’ to the left edges, then assign weight ‘1’ to the right edges.
- If you assign weight ‘1’ to the left edges, then assign weight ‘0’ to the right edges.
- Any of the above two conventions may be followed.
- But follow the same convention at the time of decoding that is adopted at the time of encoding.

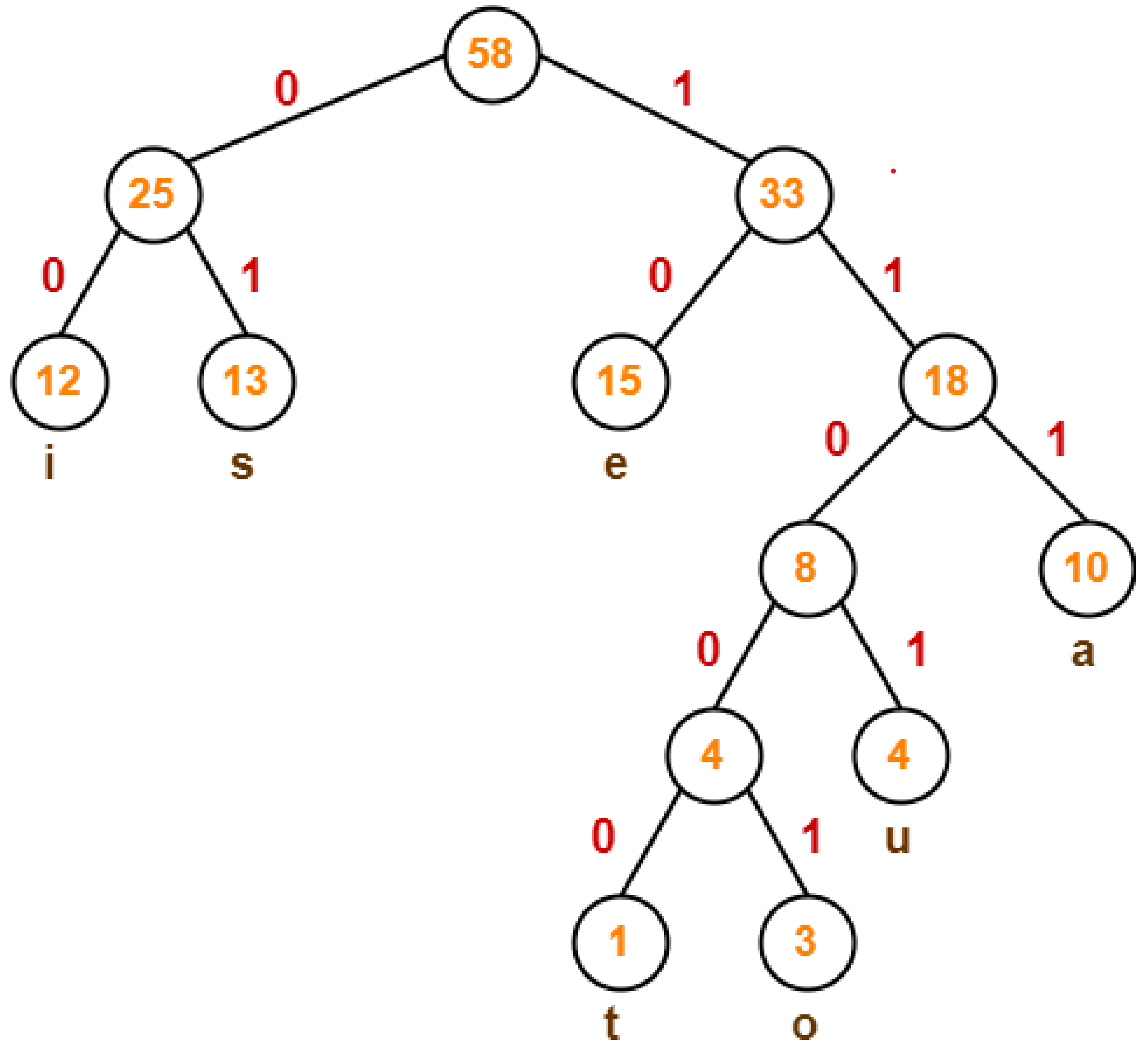


Huffman Tree

1. Huffman Code For Characters-

To write Huffman Code for any character, traverse the Huffman Tree from root node to the leaf node of that character.
Following this rule, the Huffman Code for each character is-

- a = 111
- e = 10
- i = 00
- o = 11001
- u = 1101
- s = 01
- t = 11000



Huffman Tree

Characters	Frequencies
a	10
e	15
i	12
o	3
u	4
s	13
t	1

a	= 111
e	= 10
i	= 00
o	= 11001
u	= 1101
s	= 01
t	= 11000

- Characters occurring less frequently in the text are assigned the larger code.
- Characters occurring more frequently in the text are assigned the smaller code.

2. Average Code Length-

$= \sum (\text{frequency}_i \times \text{code length}_i) / \sum (\text{frequency}_i)$

$= ((10 \times 3) + (15 \times 2) + (12 \times 2) + (3 \times 5) + (4 \times 4) + (13 \times 2) + (1 \times 5)) / (10 + 15 + 12 + 3 + 4 + 13 + 1)$

= 2.52

Characters	Frequencies
a	10
e	15
i	12
o	3
u	4
s	13
t	1
	58

a = 111
e = 10
i = 00
o = 11001
u = 1101
s = 01
t = 11000

3. Length of Huffman Encoded Message-

Using formula, total number of bits in Huffman encoded message
= Total number of characters in the message x Average code length per character
= 58 x 2.52
= 146.16
≅ 147 bits

Analysis

Time Complexity:

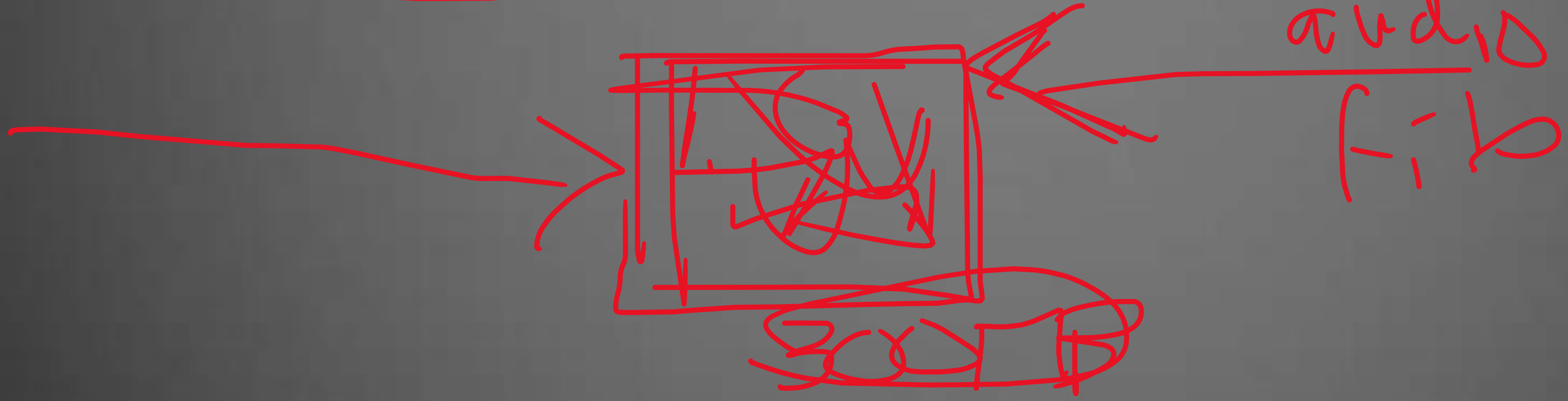
Time complexity of Huffman algorithm is O(nlogn) where each iteration requires O(logn) time to determine the cheapest weight and there would be O(n) iterations.

Applications Of Huffman Coding

Supports various file type as:

ZIP (multichannel compression including text and other data types)
JPEG
MPEG (only upto 2 layers)

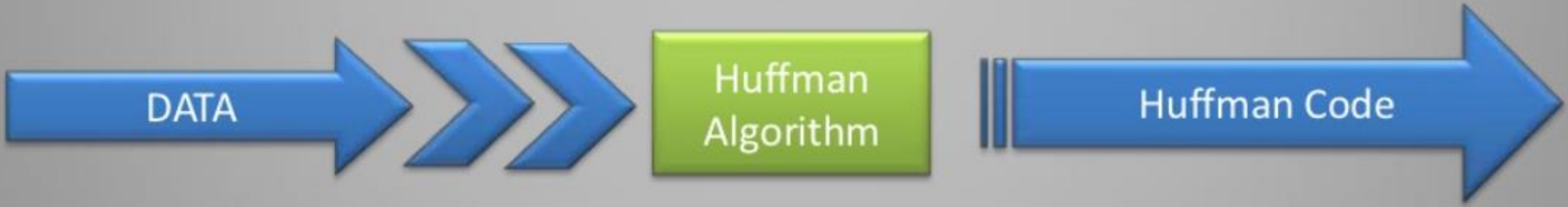
Also used in steganography for JPEG carrier compression.



Conclusion

Like many other useful algorithms we do require Huffman Algorithm for compression of data so it could be transmitted over internet and other transmission channels properly.

Huffman algorithm works on Binary trees.


DATA
Huffman Algorithm
Huffman Code