Lecture#12

Data Structures

Dr. Abu Nowshed Chy

Department of Computer Science and Engineering

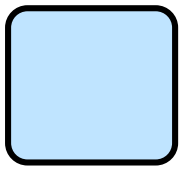
University of Chittagong

March 03, 2025

Faculty Profile

Tree

Complete Binary Tree



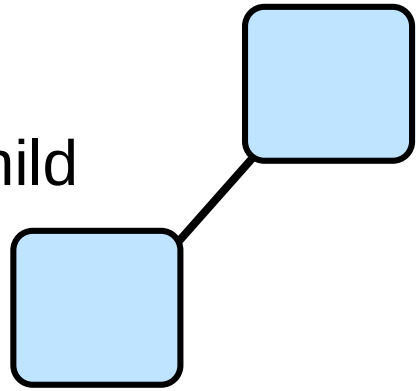
A heap is a certain kind of nearly **complete binary tree**.

- Every level except bottom is complete.
- On the bottom, nodes are placed as left as possible.

When a complete binary tree is built, its first node must be the root.

Complete Binary Tree

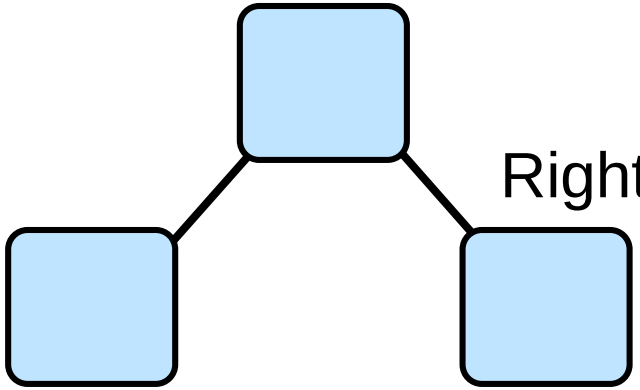
Complete binary tree.



The second node is always the left child of the root.

Complete Binary Tree

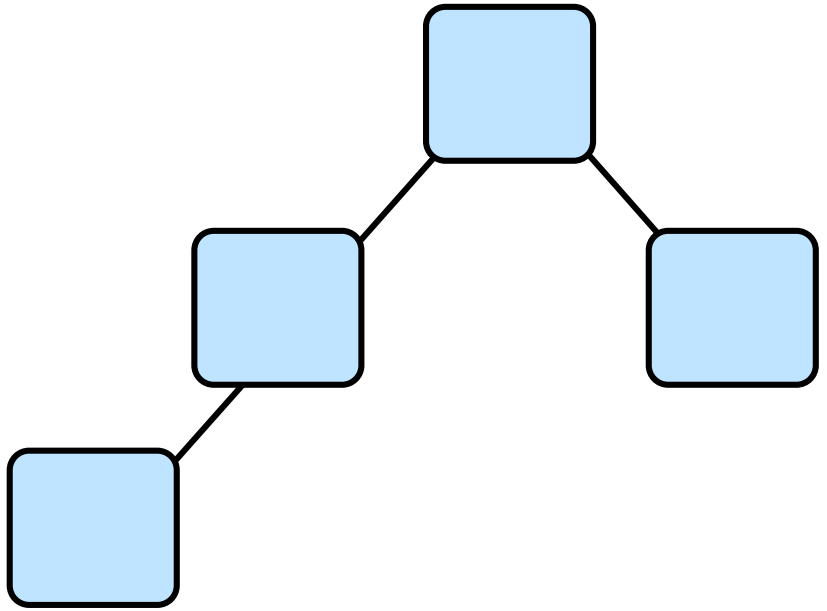
Complete binary tree.



The third node is always the right child of the root.

Complete Binary Tree

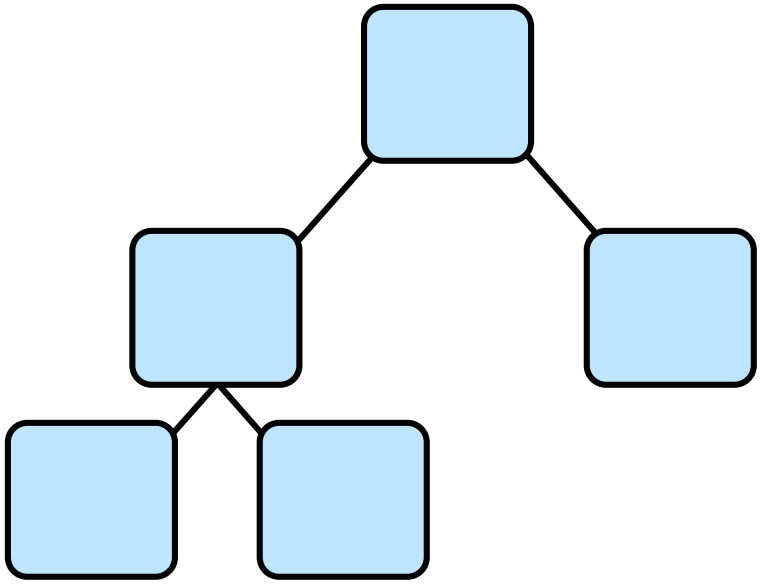
Complete binary tree.



The next nodes always fill the next level from left-to-right.

Complete Binary Tree

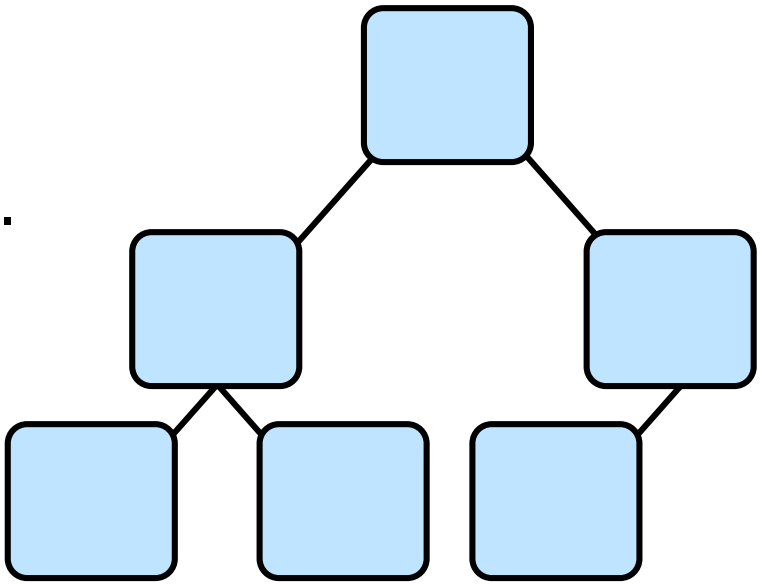
Complete binary tree.



The next nodes always fill the next level from left-to-right.

Complete Binary Tree

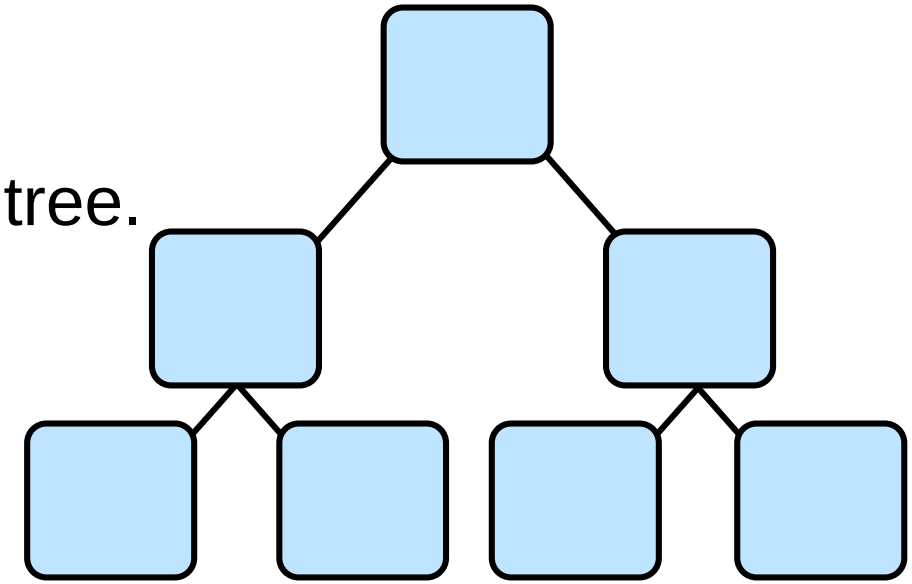
Complete binary tree.



The next nodes always fill the next level from left-to-right.

Complete Binary Tree

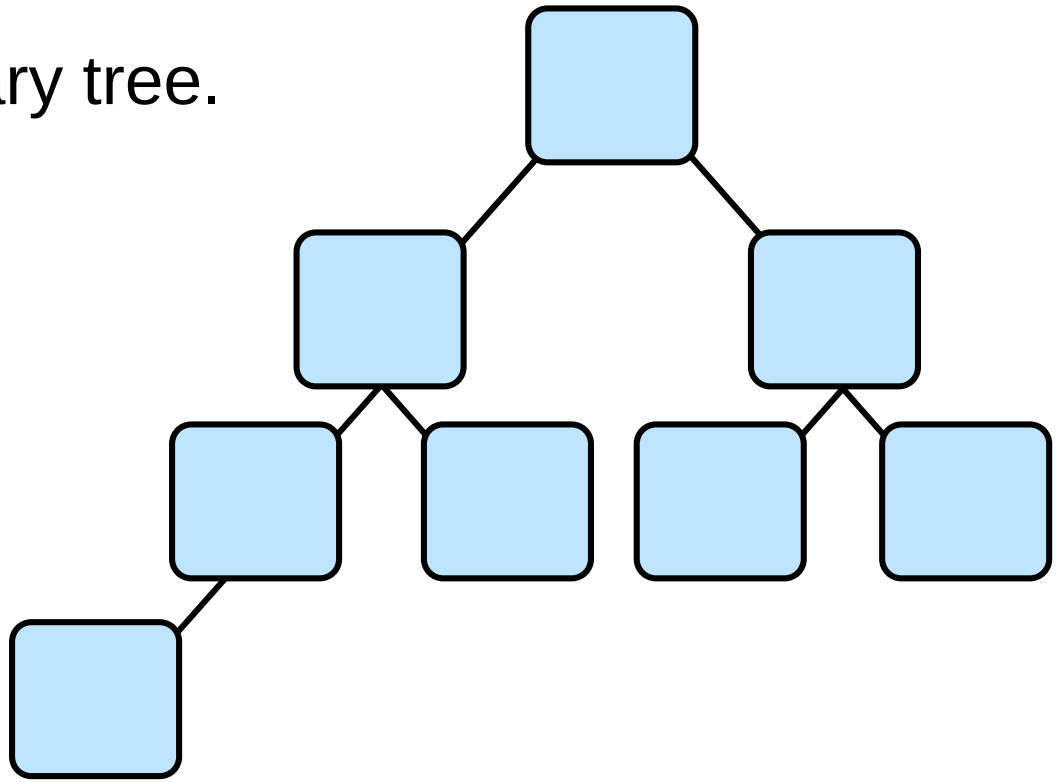
Complete binary tree.



The next nodes always fill the next level from left-to-right.

Complete Binary Tree

Complete binary tree.



Binary Search Tree (BST)

Binary Search Tree

Construct a Binary Search Tree (BST) for the following sequence of numbers-

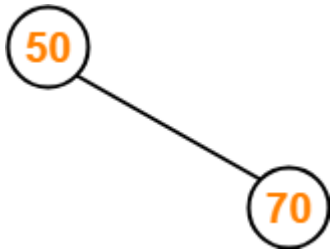
50, 70, 60, 20, 90, 10, 40, 100

Binary Search Tree

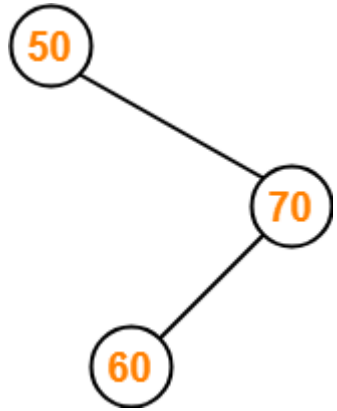
50, 70, 60, 20, 90, 10, 40, 100

Insert 50-

Insert 70-



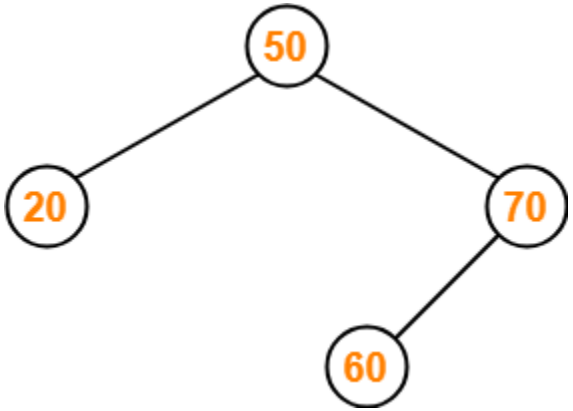
Insert 60-



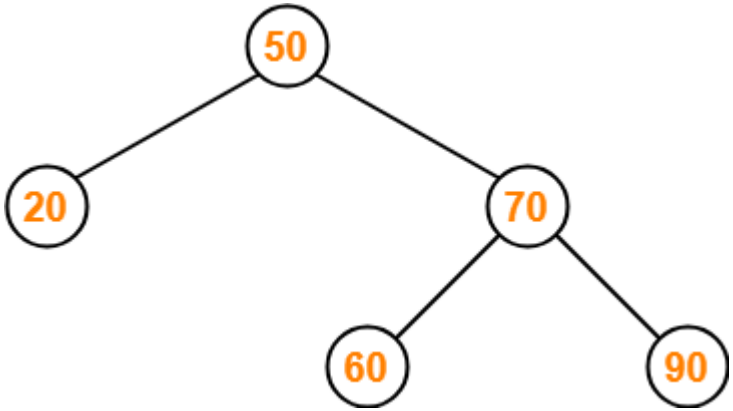
Binary Search Tree

50, 70, 60, 20, 90, 10, 40, 100

Insert 20-



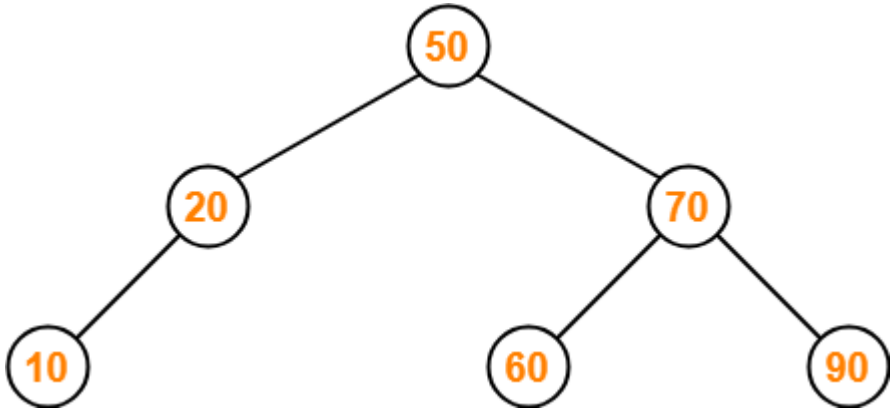
Insert 90-



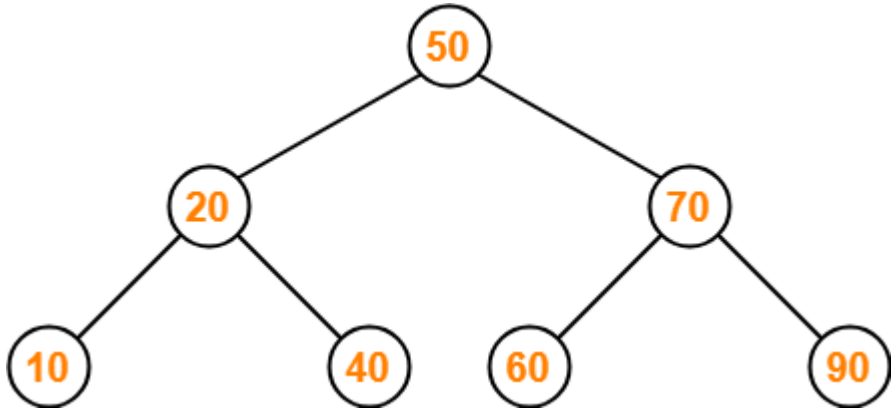
Binary Search Tree

50, 70, 60, 20, 90, 10, 40, 100

Insert 10-



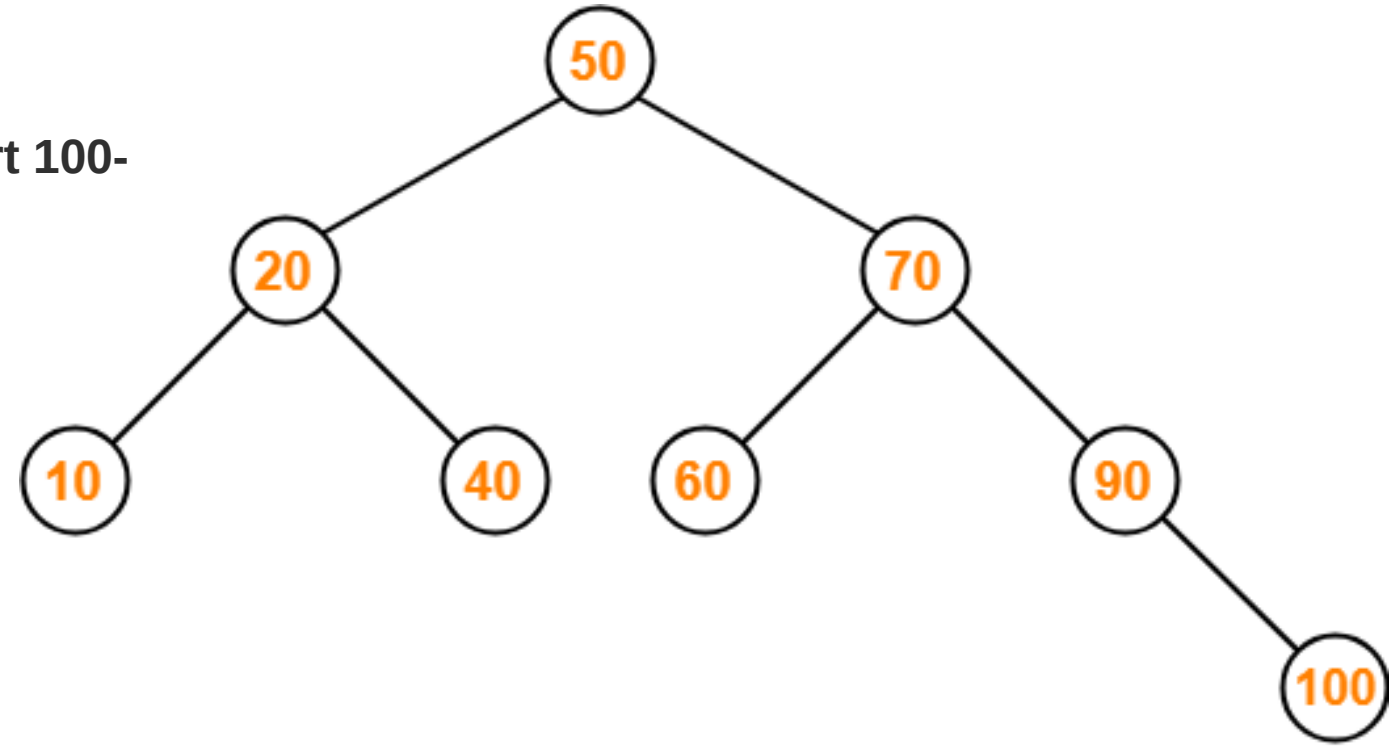
Insert 40-



Binary Search Tree

50, 70, 60, 20, 90, 10, 40, 100

Insert 100-



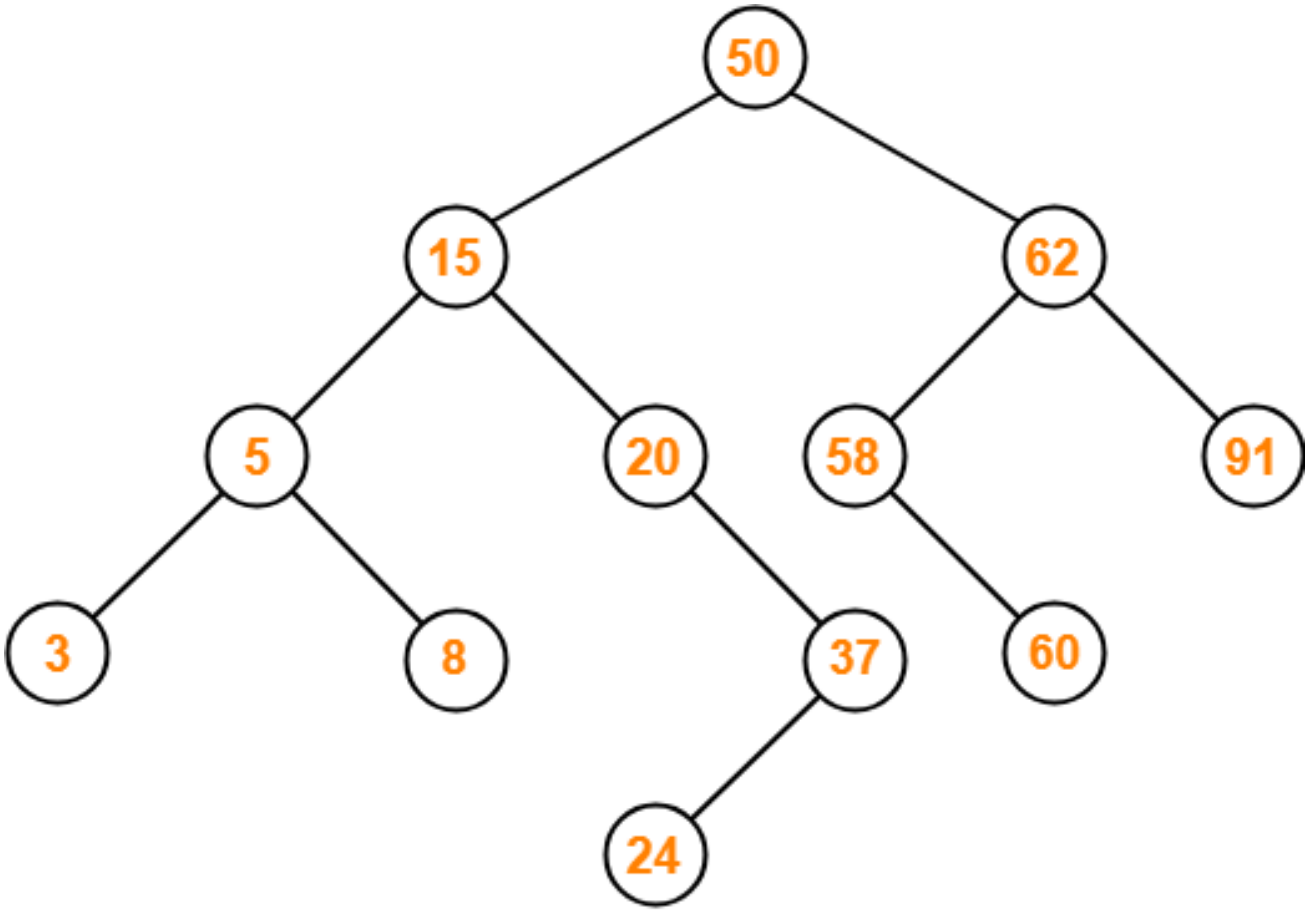
Binary Search Tree

Binary Search Tree

Construct a Binary Search Tree (BST) for the following sequence of numbers-

50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24

Binary Search Tree



Binary Search Tree

Binary Search Tree Practice

Construct a Binary Search Tree (BST) for the following sequence of numbers-

50, 76, 21, 4, 32, 64, 15, 52, 14, 100, 83, 2, 3, 70, 87, 80

Then, delete the root node from the constructed Binary Search Tree.

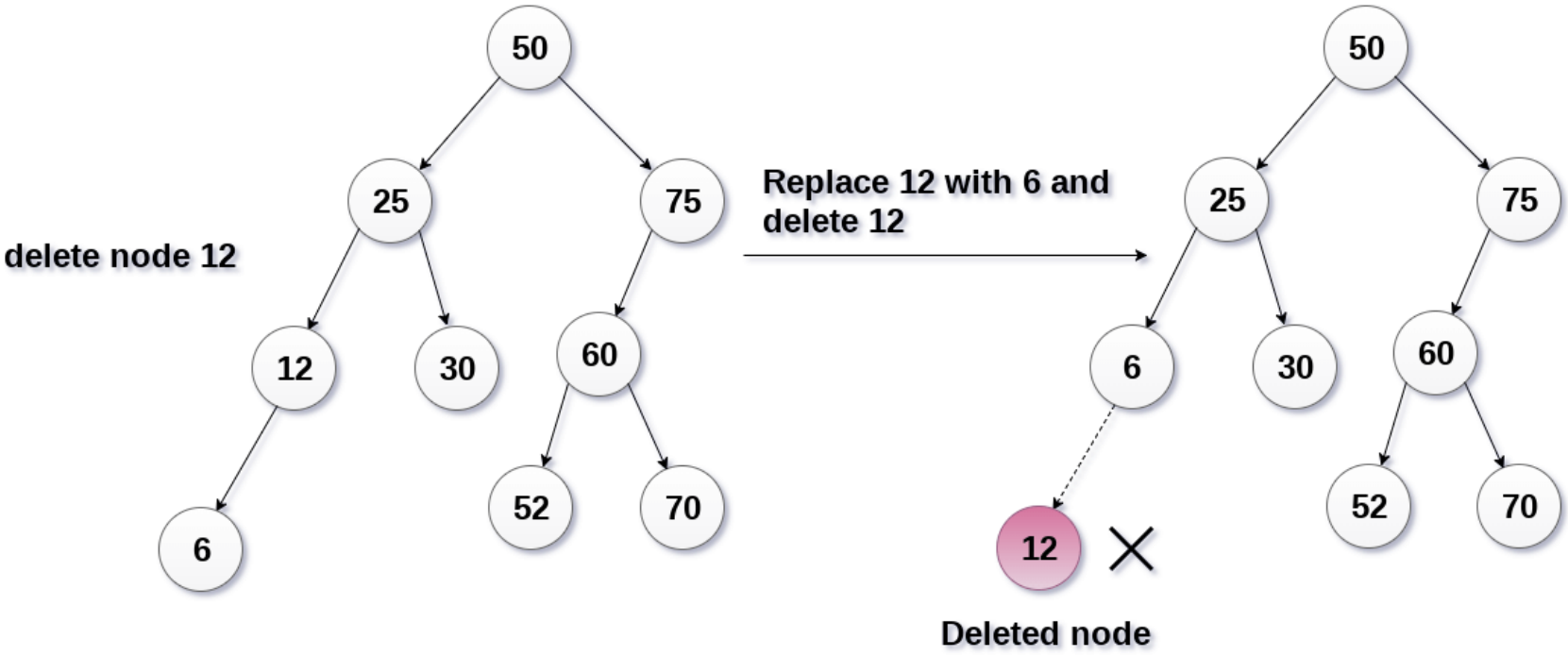
Deletion in Binary Search Tree

The node to be deleted is a leaf node

Simply remove the node from the tree.

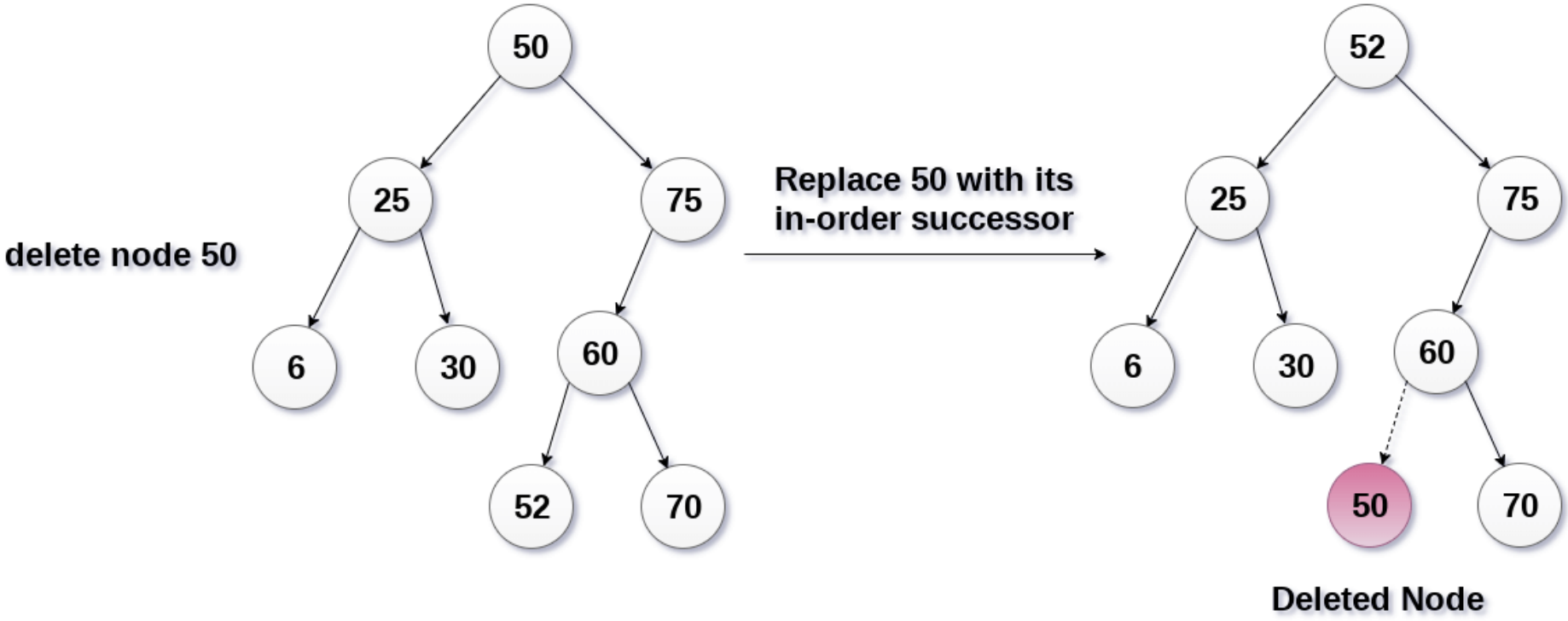
Deletion in Binary Search Tree

The node to be deleted has only one child.



Deletion in Binary Search Tree

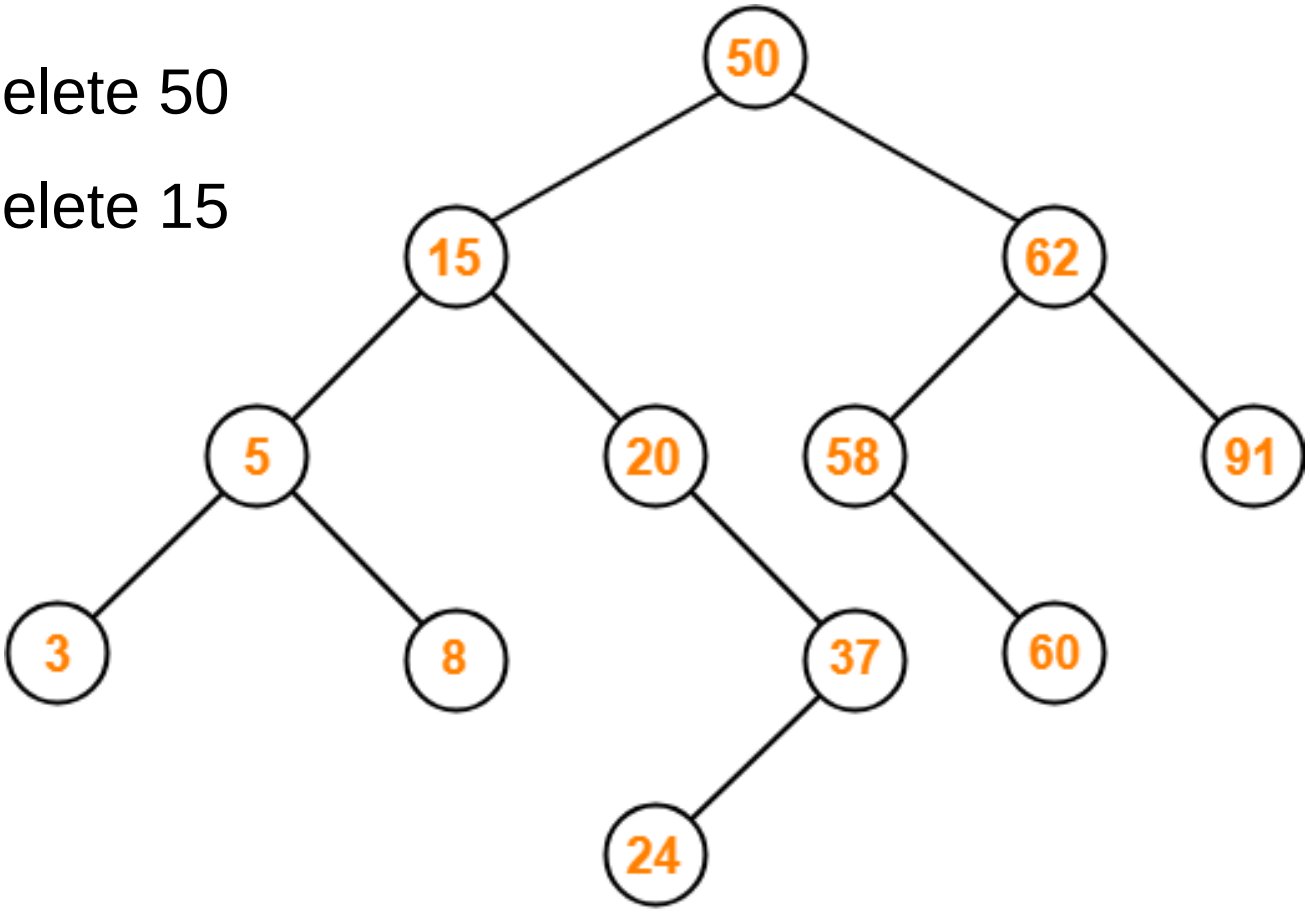
The node to be deleted has two children.



Deletion in Binary Search Tree

Delete 50

Delete 15



Binary Search Tree

Huffman Tree

Huffman Coding – Huffman Tree

The steps involved in the construction of Huffman Tree are as follows-

Step-01:

- Create a leaf node for each character of the text.
- Leaf node of a character contains the occurring frequency of that character.

Step-02:

- Arrange all the nodes in increasing order of their frequency value.

Huffman Coding – Huffman Tree

Step-03:

Considering the first two nodes having minimum frequency,

- Create a new internal node.
- The frequency of this new node is the sum of frequency of those two nodes.
- Make the first node as a left child and the other node as a right child of the newly created node.

Step-04:

- Keep repeating Step-02 and Step-03 until all the nodes form a single tree.
- The tree finally obtained is the desired Huffman Tree.

Huffman Coding – Huffman Tree

$$\text{Average code length per character} = \frac{\Sigma\ (\text{frequency}_i \times \text{code length}_i)}{\Sigma\ \text{frequency}_i}$$

Total number of bits in Huffman encoded message

= Total number of characters in the message x
Average code length per character

Huffman Coding – Huffman Tree

Problem-

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

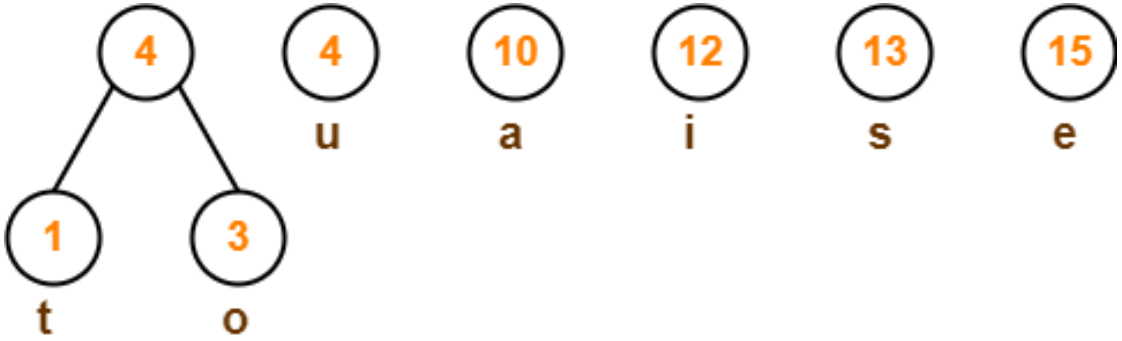
- Huffman Code for each character
- Average code length
- Length of Huffman encoded message (in bits)

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

Huffman Coding – Huffman Tree

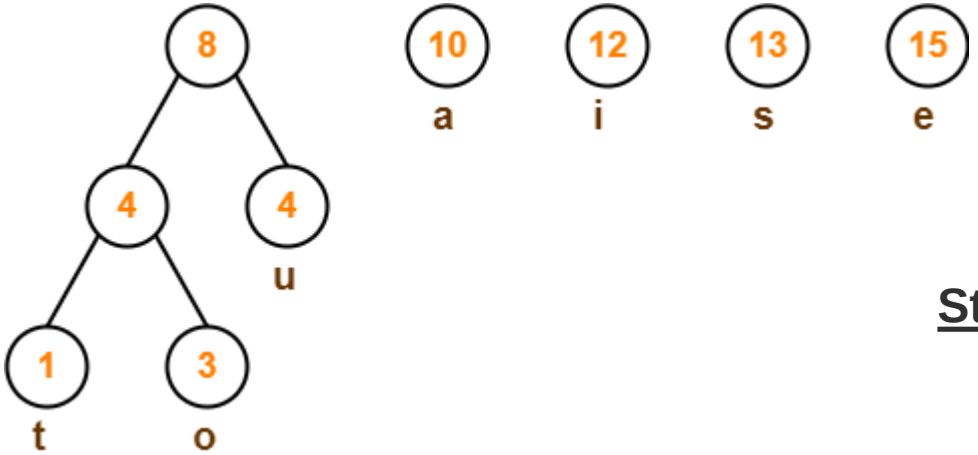
Step-01:

Step-02:

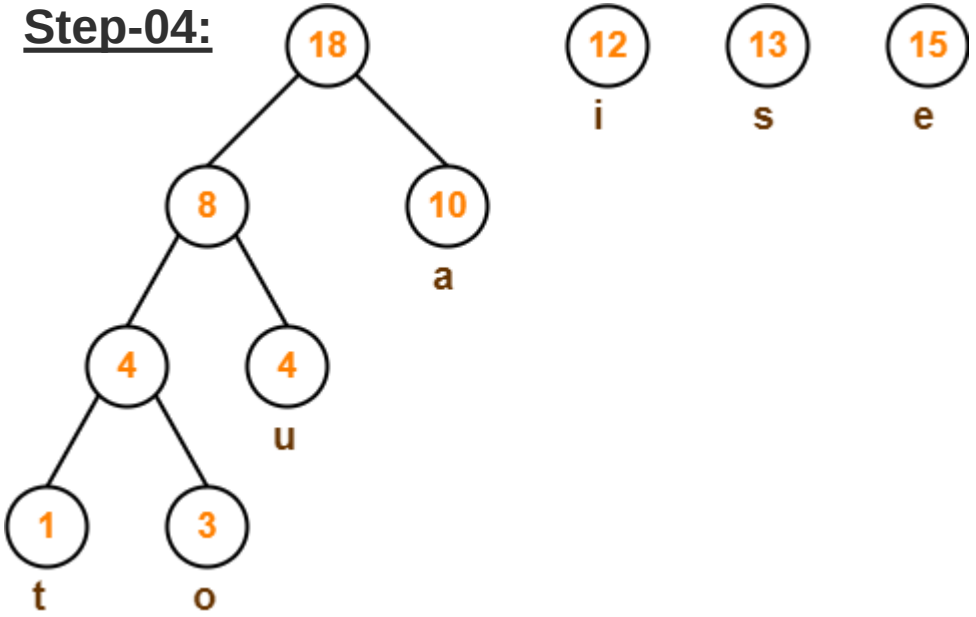


Huffman Coding – Huffman Tree

Step-03:

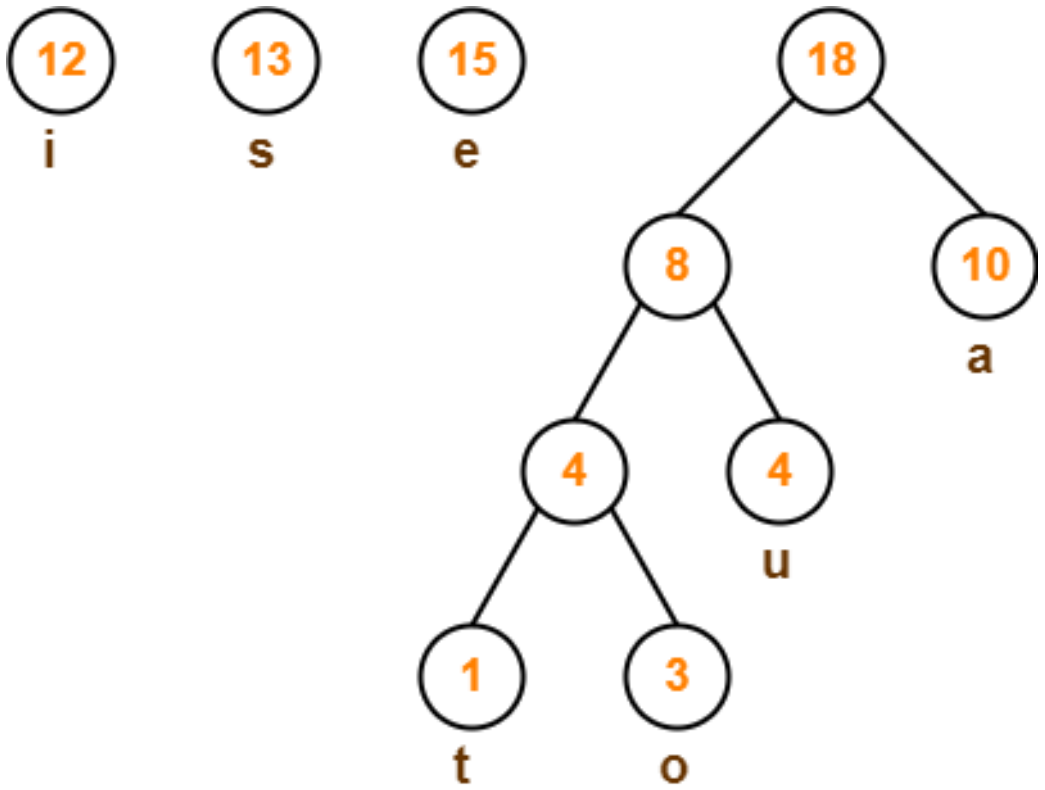


Step-04:

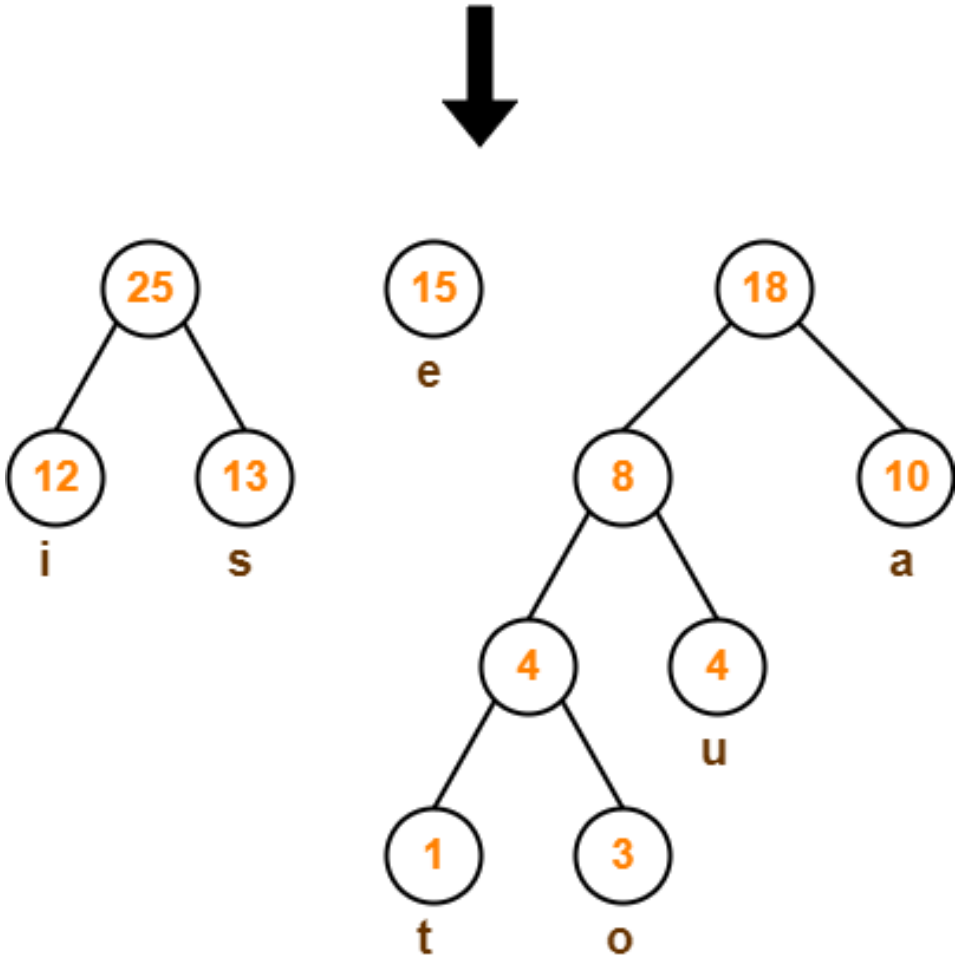


Huffman Coding – Huffman Tree

Step-05:

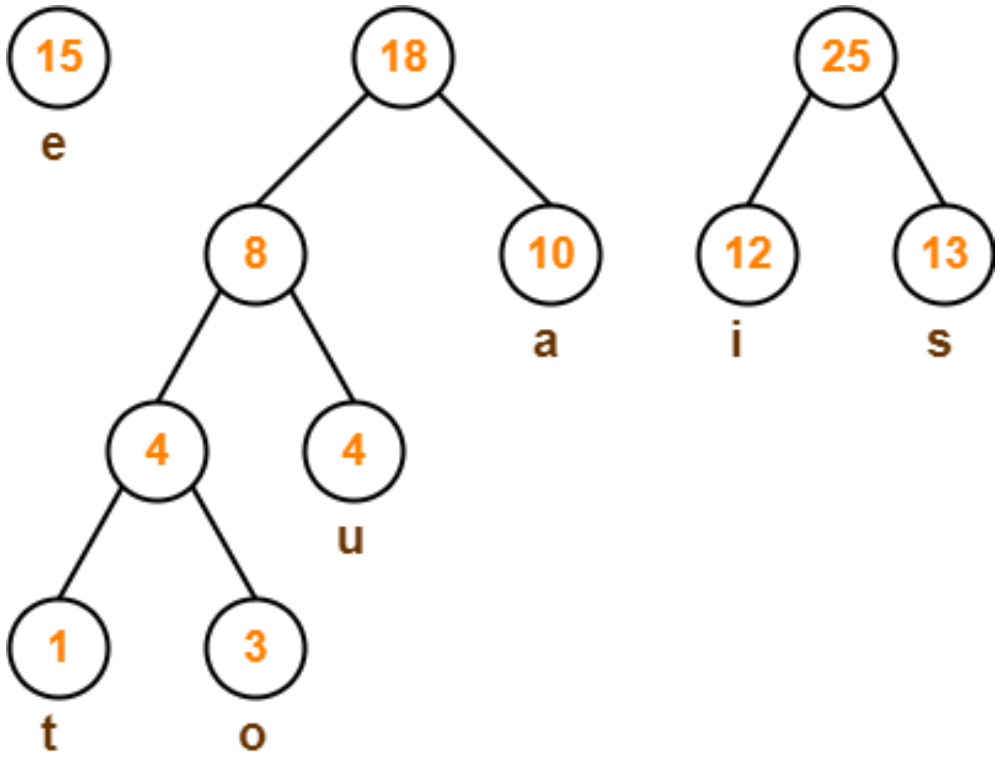


Huffman Coding – Huffman Tree

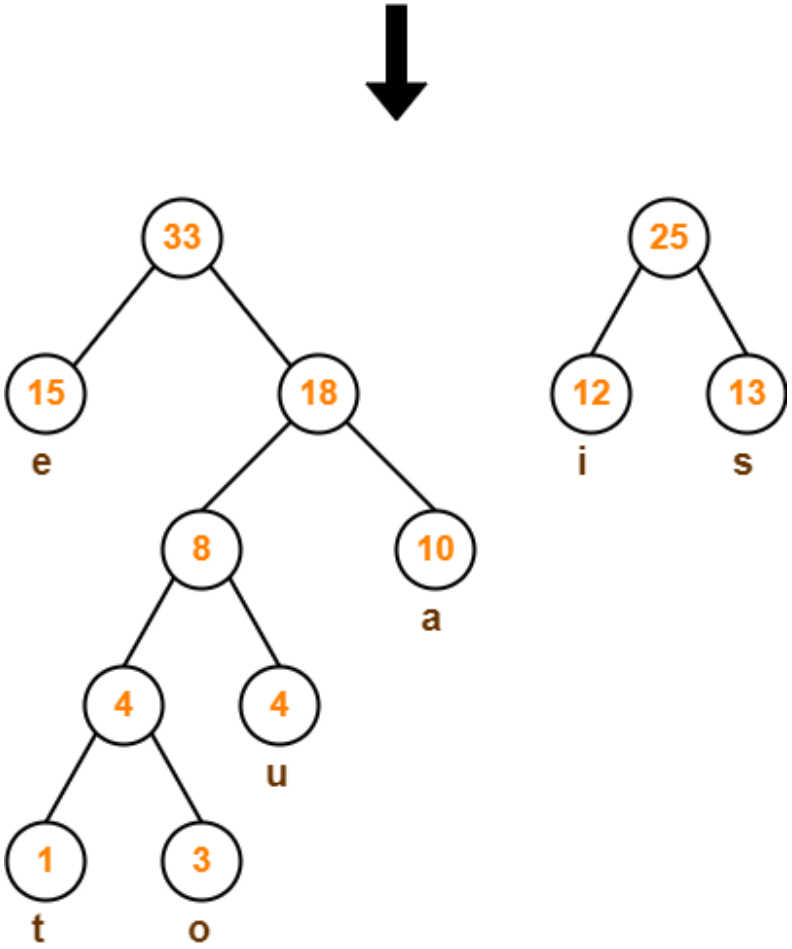


Huffman Coding – Huffman Tree

Step-06:

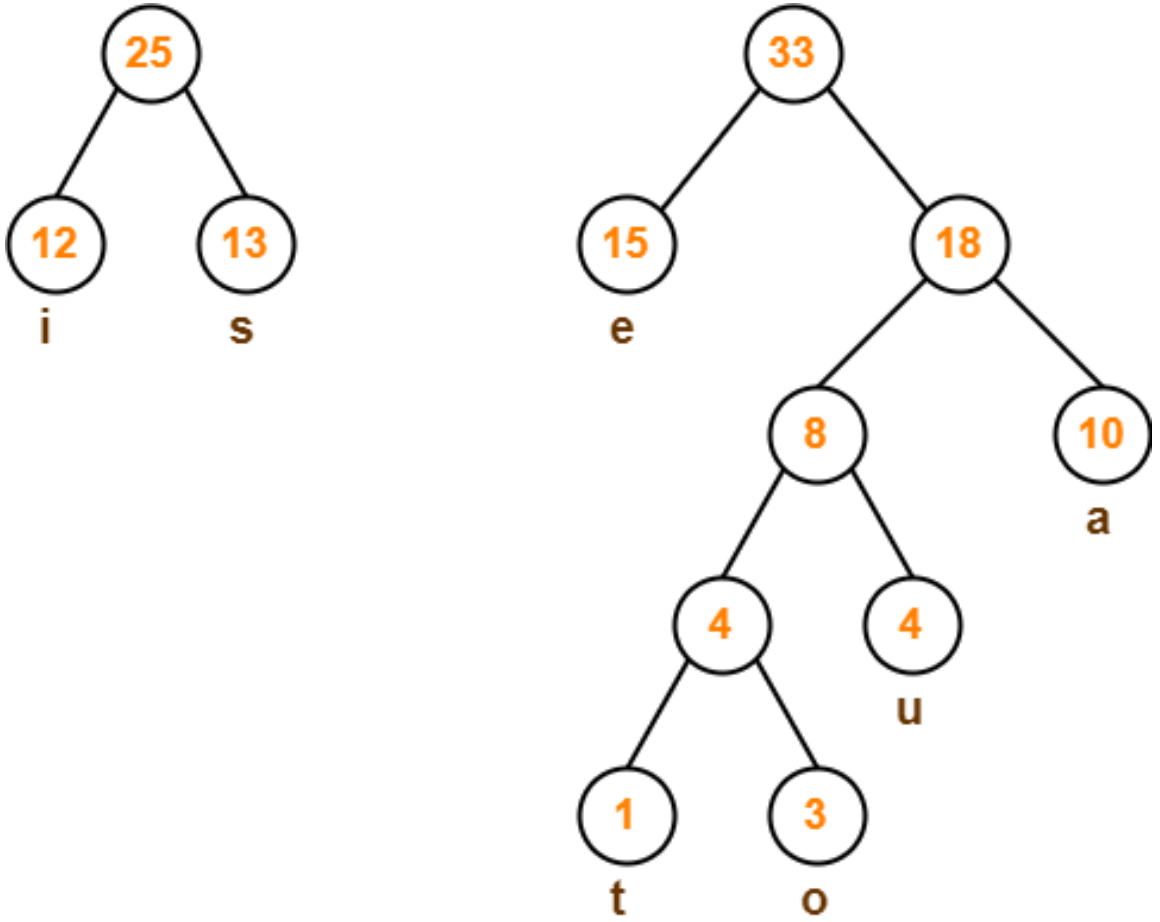


Huffman Coding – Huffman Tree

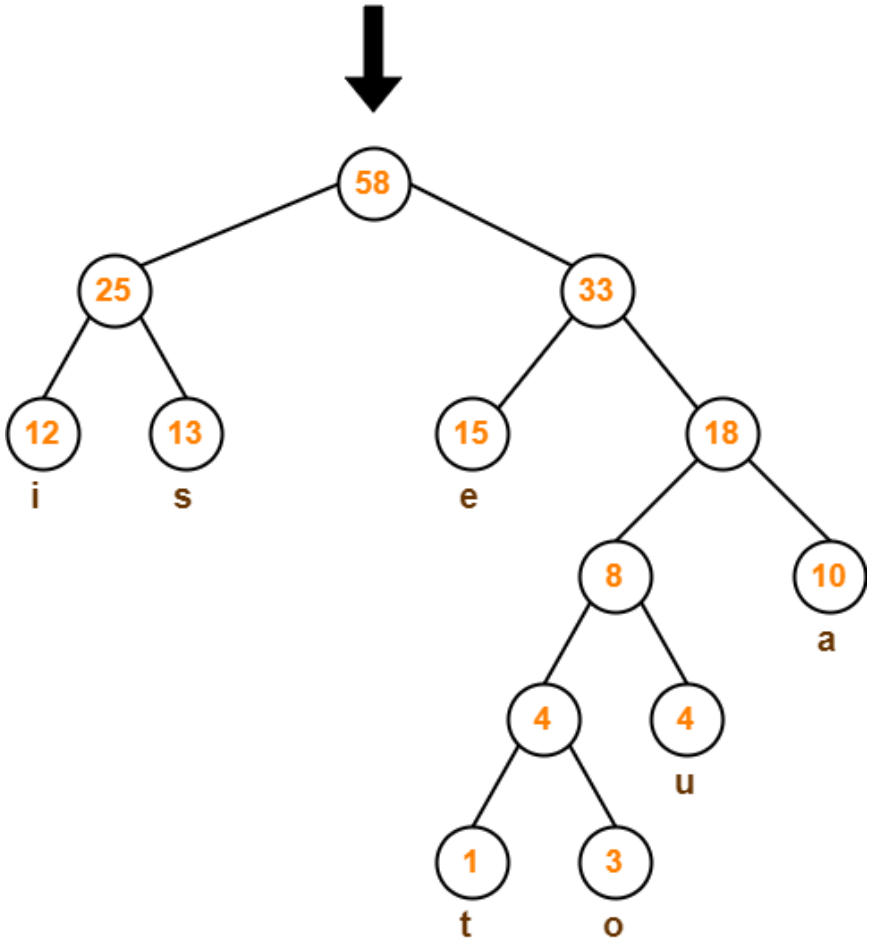


Huffman Coding – Huffman Tree

Step-07:



Huffman Coding – Huffman Tree



Huffman Tree

Huffman Coding – Huffman Tree

After assigning weight to all the edges, the modified Huffman Tree is-:

To write Huffman Code for any character, traverse the Huffman Tree from root node to the leaf node of that character.

a = 111

e = 10

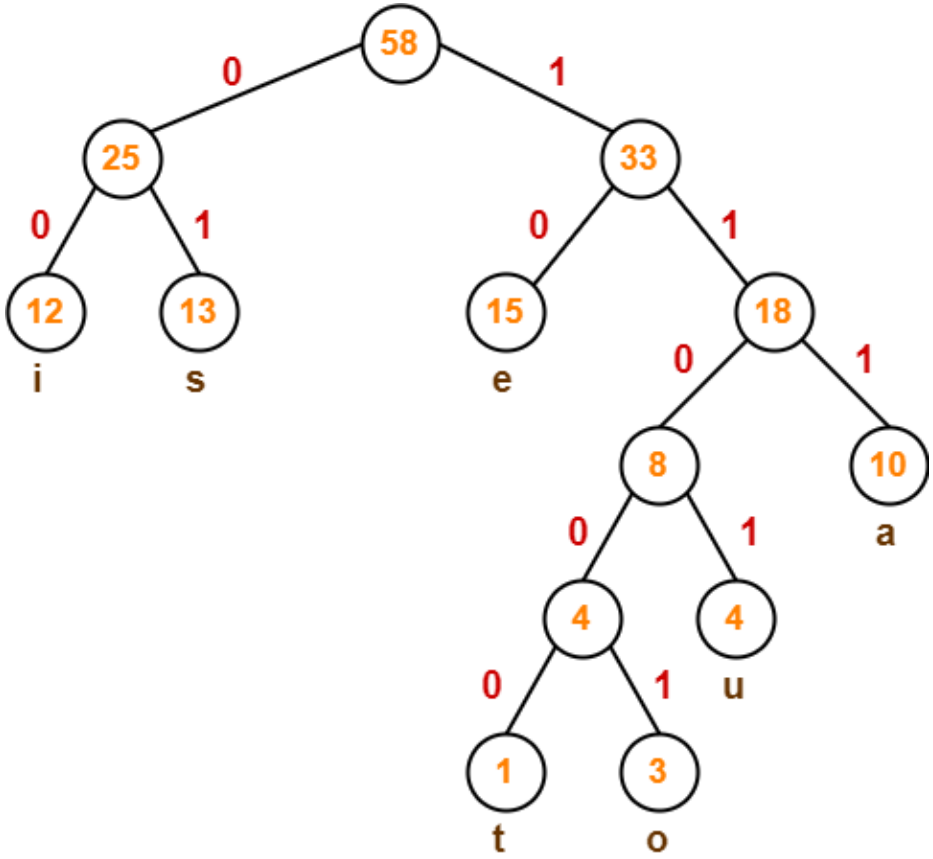
i = 00

o = 11001

u = 1101

s = 01

t = 11000



Huffman Tree

Huffman Coding – Huffman Tree

Characters	Frequencies	Code	Code Length	Total Length
a	10	111	3	30
e	15	10	2	30
i	12	00	2	24
o	3	11001	5	15
u	4	1101	4	16
s	13	01	2	26
t	1	11000	5	5

146 bits

Huffman Coding – Huffman Tree

Average code length:

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

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

$= 2.52$

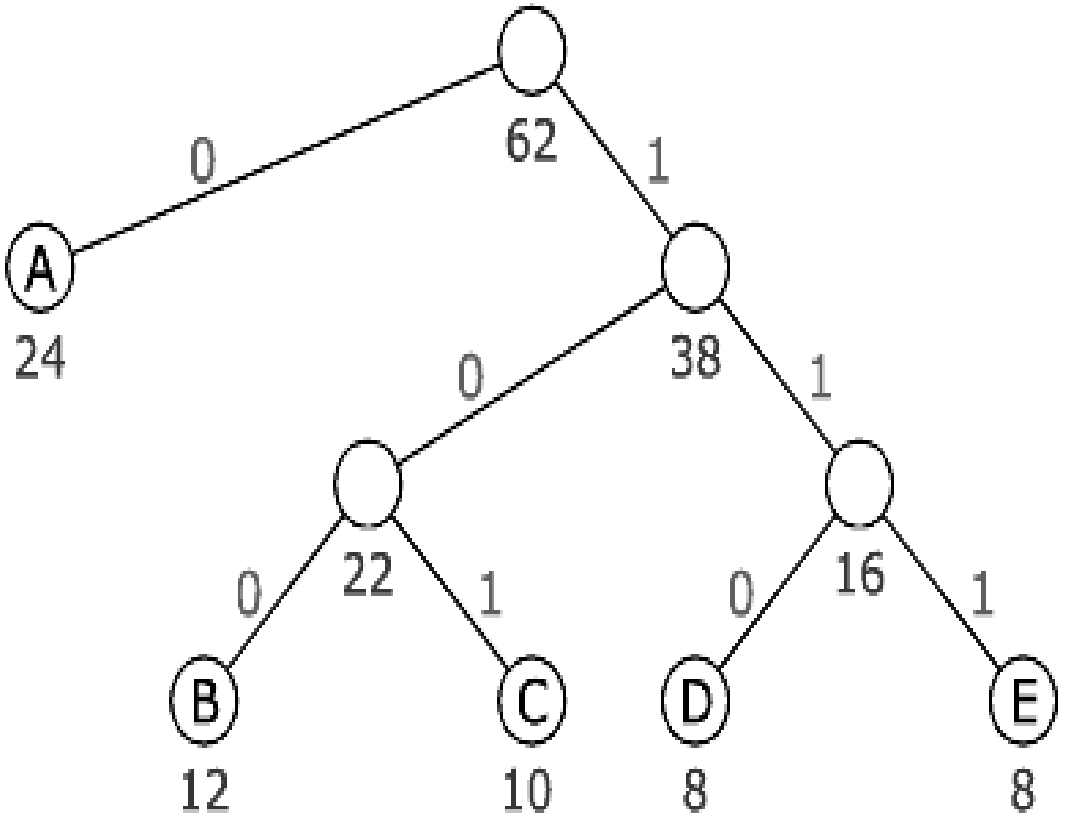
Huffman Coding – Huffman Tree

Problem- A file contains the following characters with the frequencies as shown. determine-

- Huffman Code for each character
- Average code length
- Length of Huffman encoded message (in bits)

Symbol	Frequency
A	24
B	12
C	10
D	8
E	8

Huffman Coding – Huffman Tree



Huffman Coding – Huffman Tree

Symbol	Frequency	Code	Code Length	Total Length
A	24	0	1	24
B	12	100	3	36
C	10	101	3	30
D	8	110	3	24
E	8	111	3	24

Total : 138 bits


Thank You!