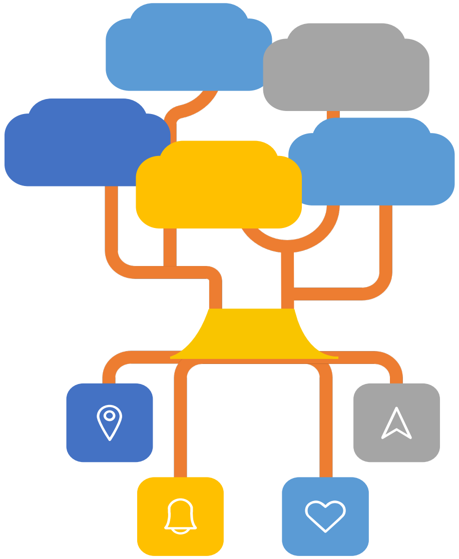
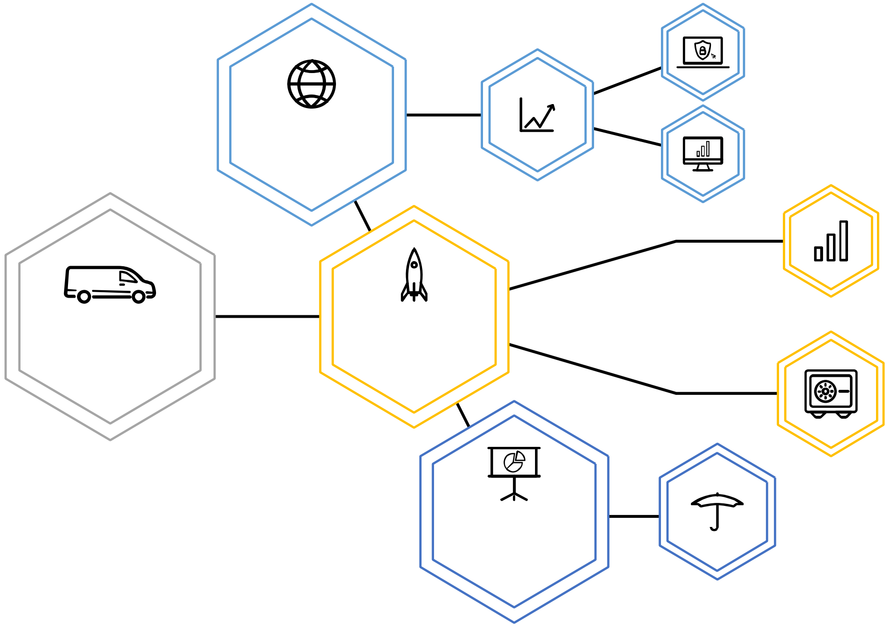




AV121: Data Structures and Algorithms

Tutorial-03

Tutorial 03 – Plan

- Binary Trees
- Binary Search Tree
- AVL Trees
  - → Properties, Insertion
- Red-black Trees
  - → Properties, Insertion

Binary Search Trees

- Which of the following statements is true
  A. In-order traversal of a binary search tree containing numbers gives the sorted list of numbers as output
  B. A binary search tree can be re-constructed, given its preorder traversal alone
  C. Both A&B
  D. None of the above

Binary Search Trees

- Which of the following statements is true
  A. All AVL Trees are binary search trees
  B. All Red-Black Trees are binary search trees
  C. All binary search trees are AVL Trees
  D. Both A&B
  E. All of the above

Binary Search Trees

- Search operation in a binary search tree always completes in O(log n) time
  - A. True
  - B. False
  - C. Can’t say

Binary Search Trees

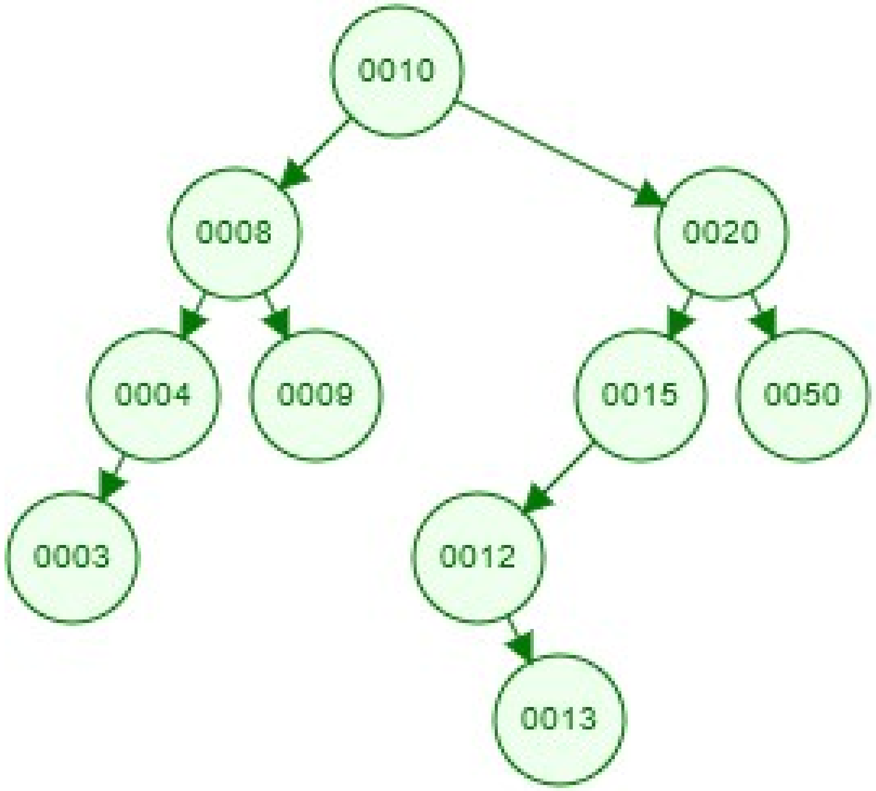
- Search operation in an AVL tree always completes in O(log n) time
  A. True
  B. False
  C. Can’t say

Binary Search Trees

- Search operation in a Red-Black tree always completes in O(log n) time
  A. True
  B. False
  C. Can’t say

Binary Search Trees

- Delete 10 from the BST below

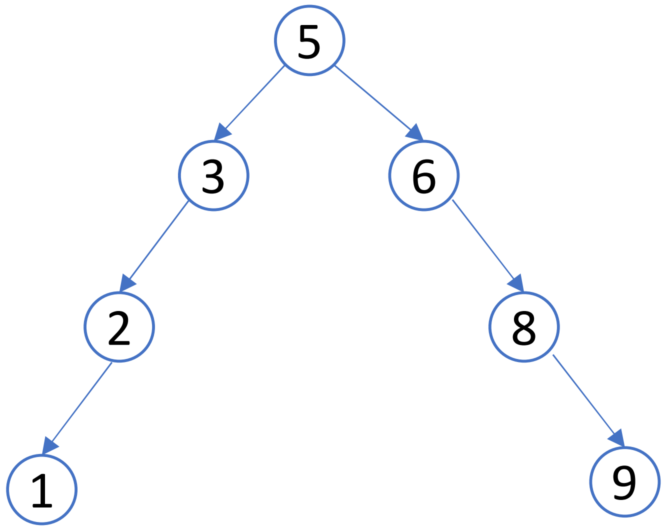


Binary Search Trees

- In an AVL Tree, for every node in the tree, height of the left and right sub-trees can differ by at most 1
  - A. True
  - B. False
  - C. Can’t say

Binary Search Trees

- Is this an AVL Tree?

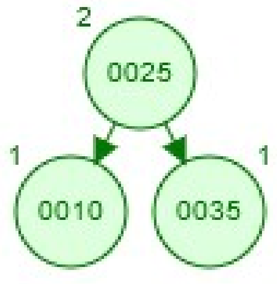


Binary Search Trees

- Insert the following numbers in to an AVL Tree
  - → 10, 25, 35, 45, 55, 75, 5, 4, 3, 2, 1

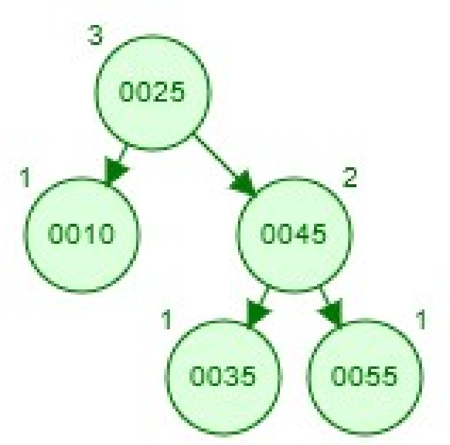
Binary Search Trees

- Insert the following numbers in to an AVL Tree
  - → 10, 25, 35, 45, 55, 75, 5, 4, 3, 2, 1



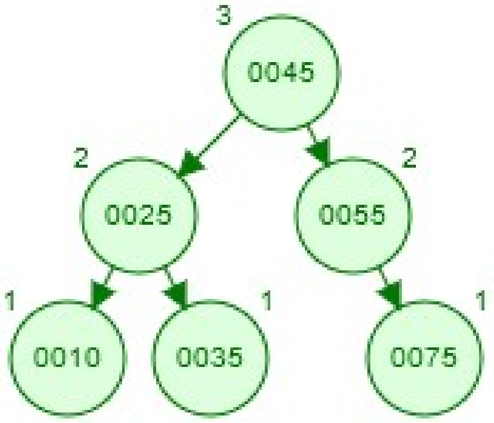
Binary Search Trees

- Insert the following numbers in to an AVL Tree
  - → 10, 25, 35, 45, 55, 75, 5, 4, 3, 2, 1



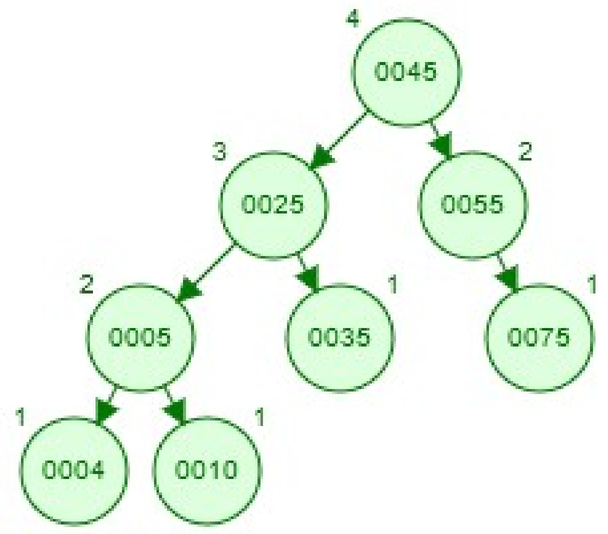
Binary Search Trees

- Insert the following numbers in to an AVL Tree
  - → 10, 25, 35, 45, 55, 75, 5, 4, 3, 2, 1



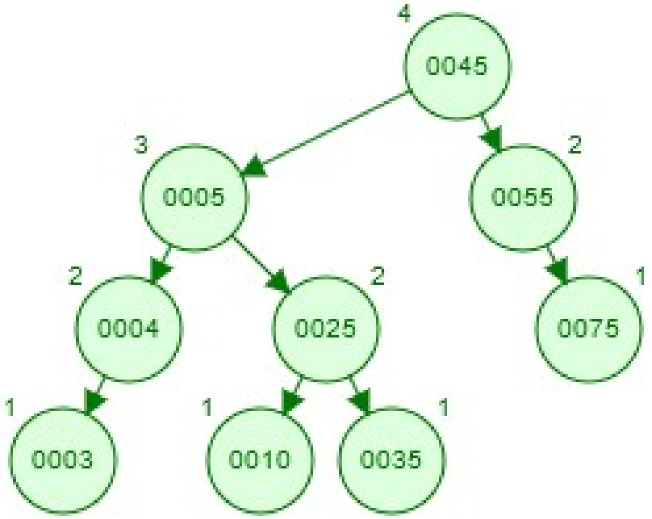
Binary Search Trees

- Insert the following numbers in to an AVL Tree
  - → 10, 25, 35, 45, 55, 75, 5, 4, 3, 2, 1



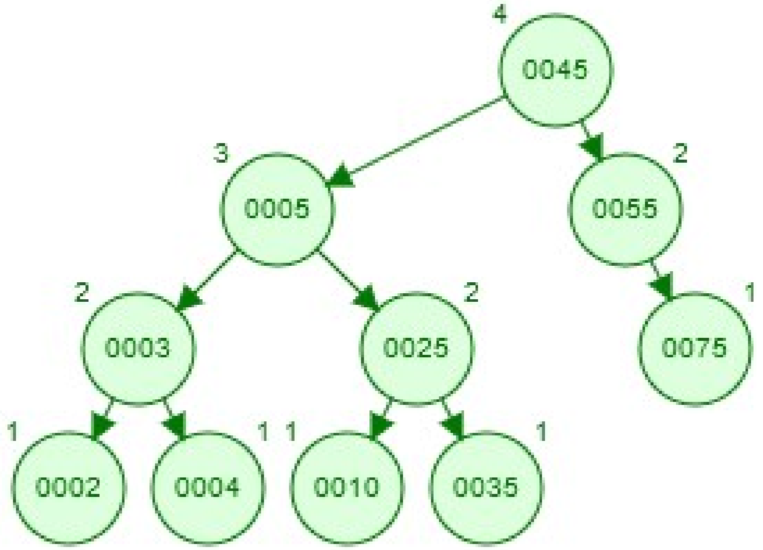
Binary Search Trees

- Insert the following numbers in to an AVL Tree
  - → 10, 25, 35, 45, 55, 75, 5, 4, 3, 2, 1



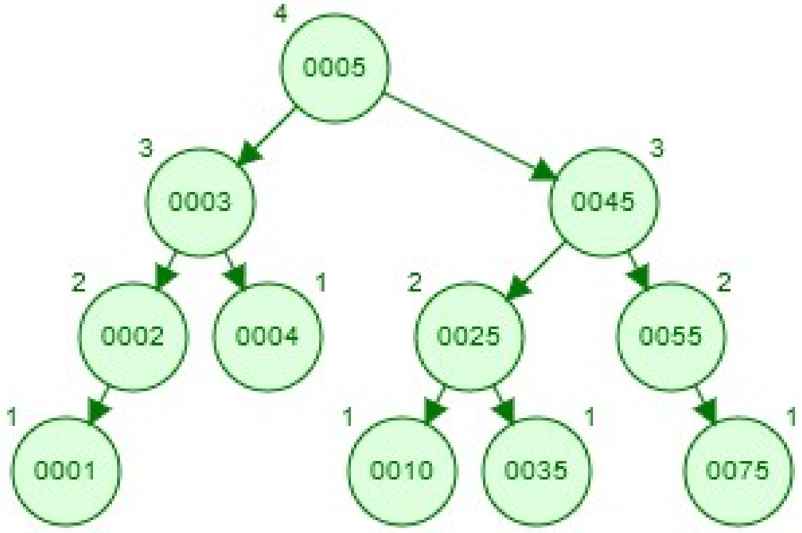
Binary Search Trees

- Insert the following numbers in to an AVL Tree
  - → 10, 25, 35, 45, 55, 75, 5, 4, 3, 2, 1



Binary Search Trees

- Insert the following numbers in to an AVL Tree
  - → 10, 25, 35, 45, 55, 75, 5, 4, 3, 2, 1



Binary Search Trees

- Which of the following statements about Red-Black trees is false
  A. Every node is either red or black
  B. A red node can have only red children
  C. A black node cannot have black children
  D. Every path from the root to any leaf node must contain the same number of black nodes

Binary Search Trees

- In an RB Tree, path from the root to the farthest leaf is no more than twice as long as the path from the root to the nearest leaf
  A. True
  B. False
  C. Can’t say

Binary Search Trees

- In RB Trees, balance is maintained by
  A. Doing tree rotations
  B. Doing color manipulation
  C. Both A&B
  D. None of the above

Huffman Coding

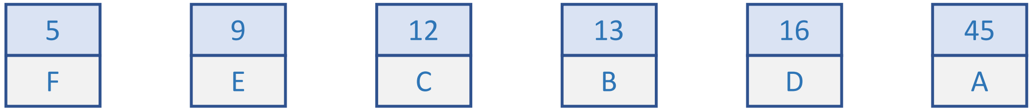
- Construct the Huffman Tree for the character set given below:

Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5

Huffman Coding

- Construct the Huffman Tree for the character set given below:

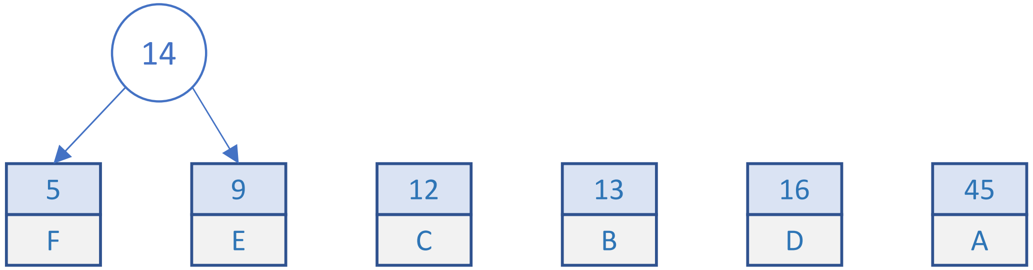
Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5



Huffman Coding

- Construct the Huffman Tree for the character set given below:

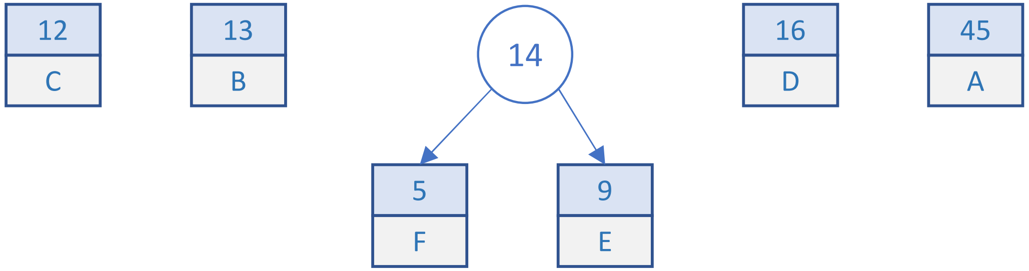
Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5



Huffman Coding

- Construct the Huffman Tree for the character set given below:

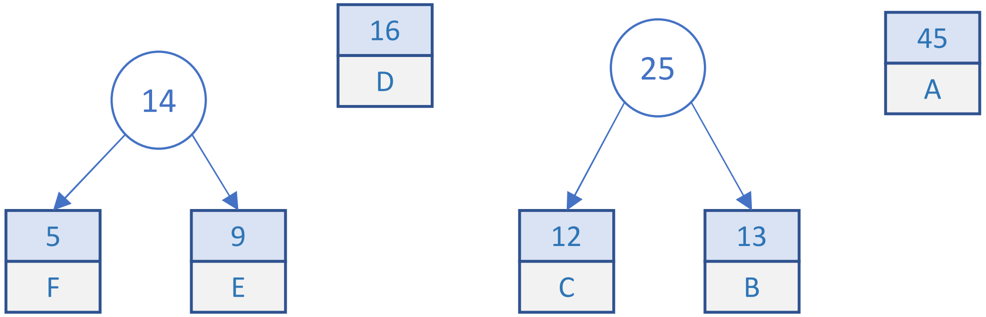
Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5



Huffman Coding

- Construct the Huffman Tree for the character set given below:

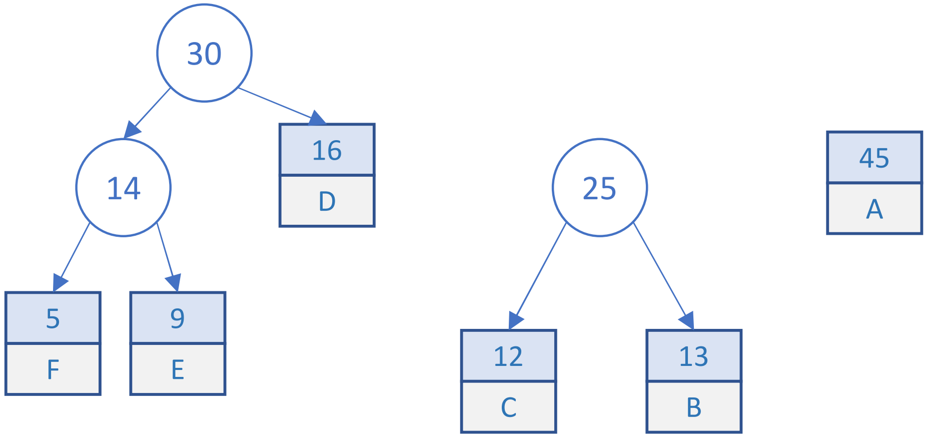
Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5



Huffman Coding

- Construct the Huffman Tree for the character set given below:

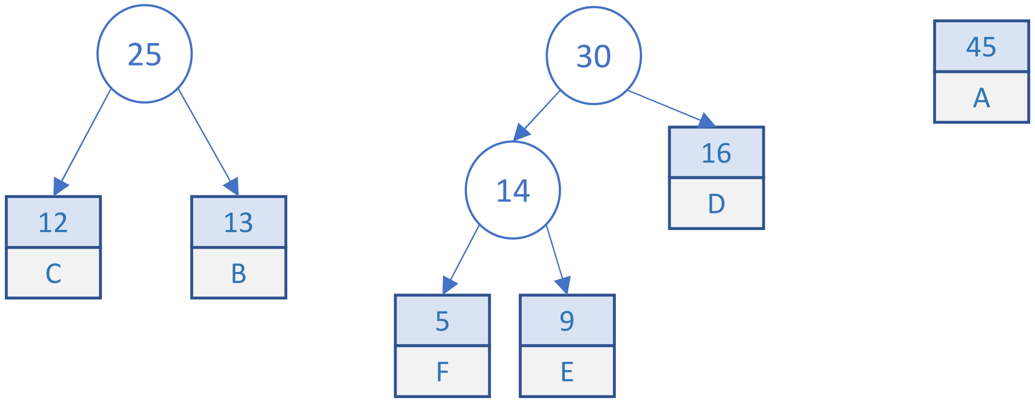
Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5



Huffman Coding

- Construct the Huffman Tree for the character set given below:

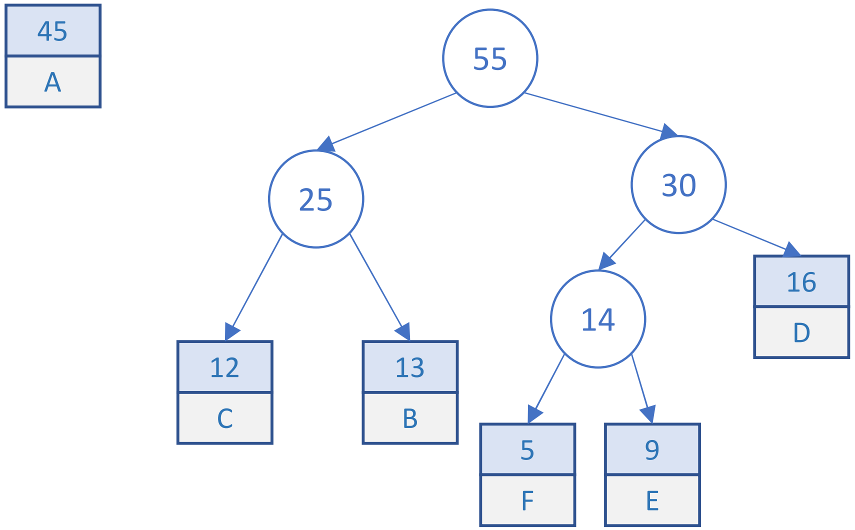
Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5



Huffman Coding

- Construct the Huffman Tree for the character set given below:

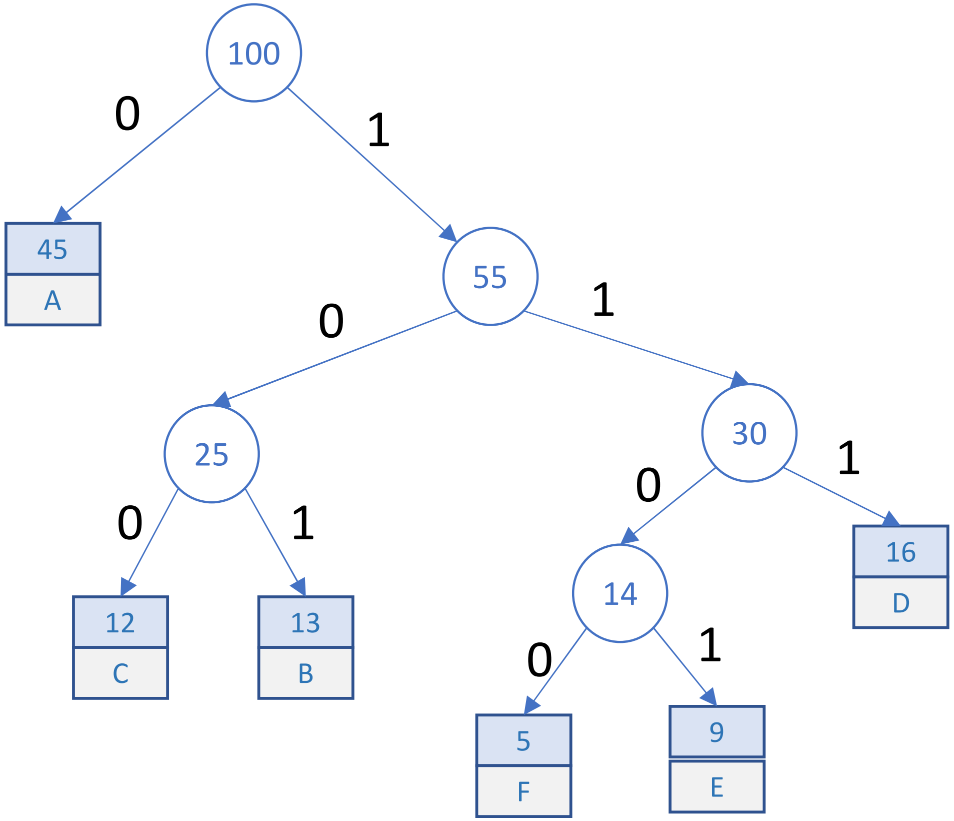
Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5



Huffman Coding

- Construct the Huffman Tree for the character set given below:

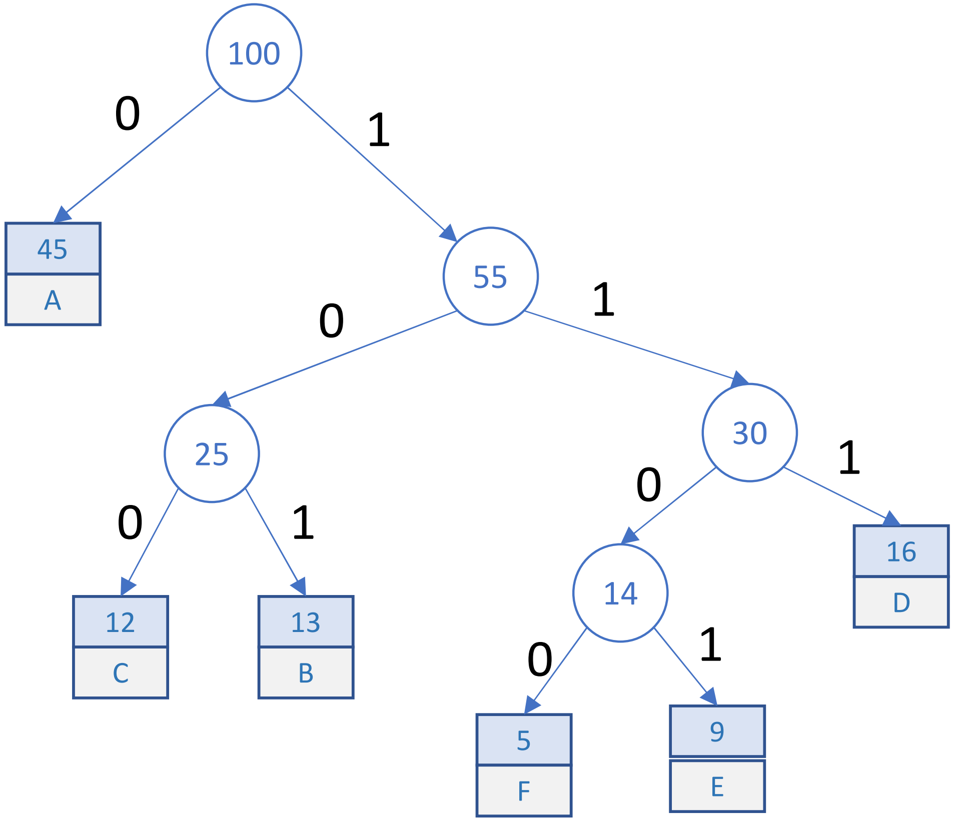
Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5



Huffman Coding

- Construct the Huffman Tree for the character set given below:

Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5
Code	0	101	100	111	1101	1100

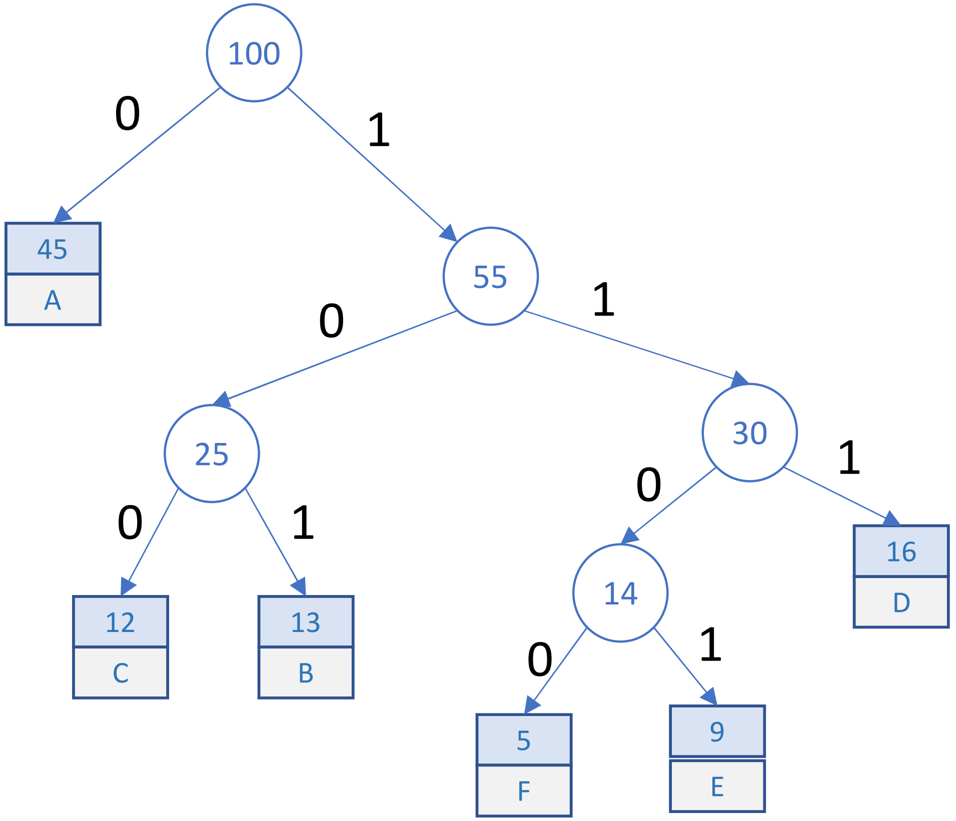


Decode 001011101

Huffman Coding

- Construct the Huffman Tree for the character set given below:

Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5
Code	0	101	100	111	1101	1100



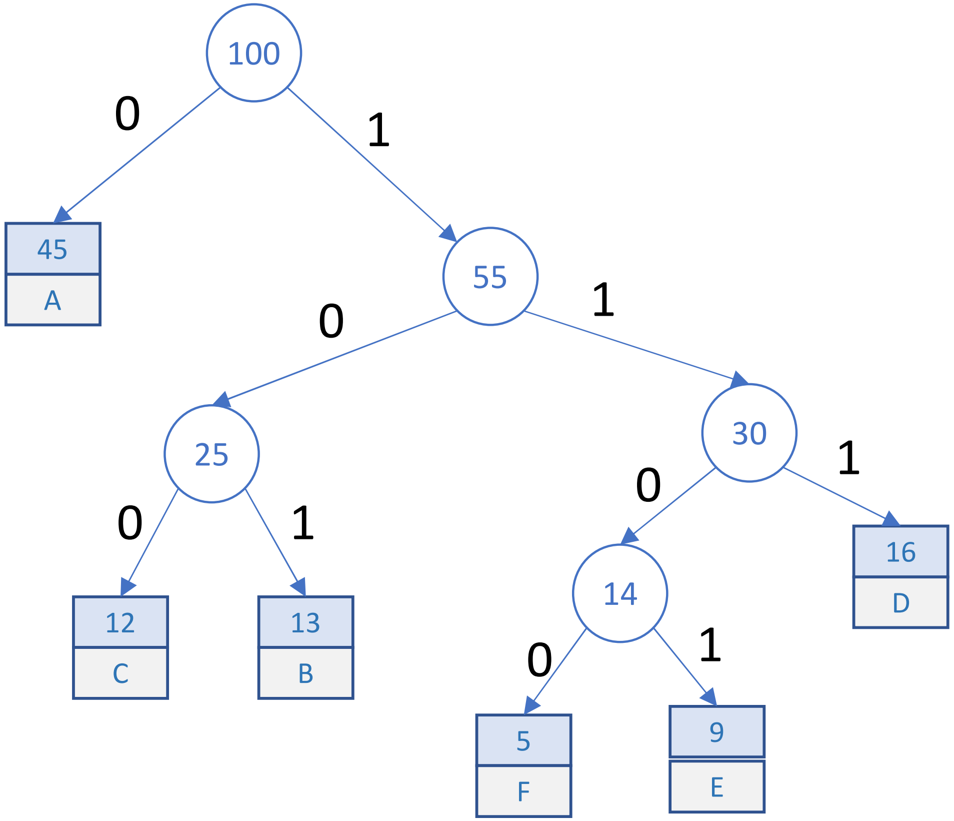
0 0 101 1101

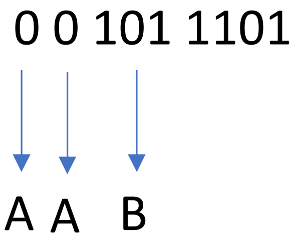
A A

Huffman Coding

- Construct the Huffman Tree for the character set given below:

Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5
Code	0	101	100	111	1101	1100



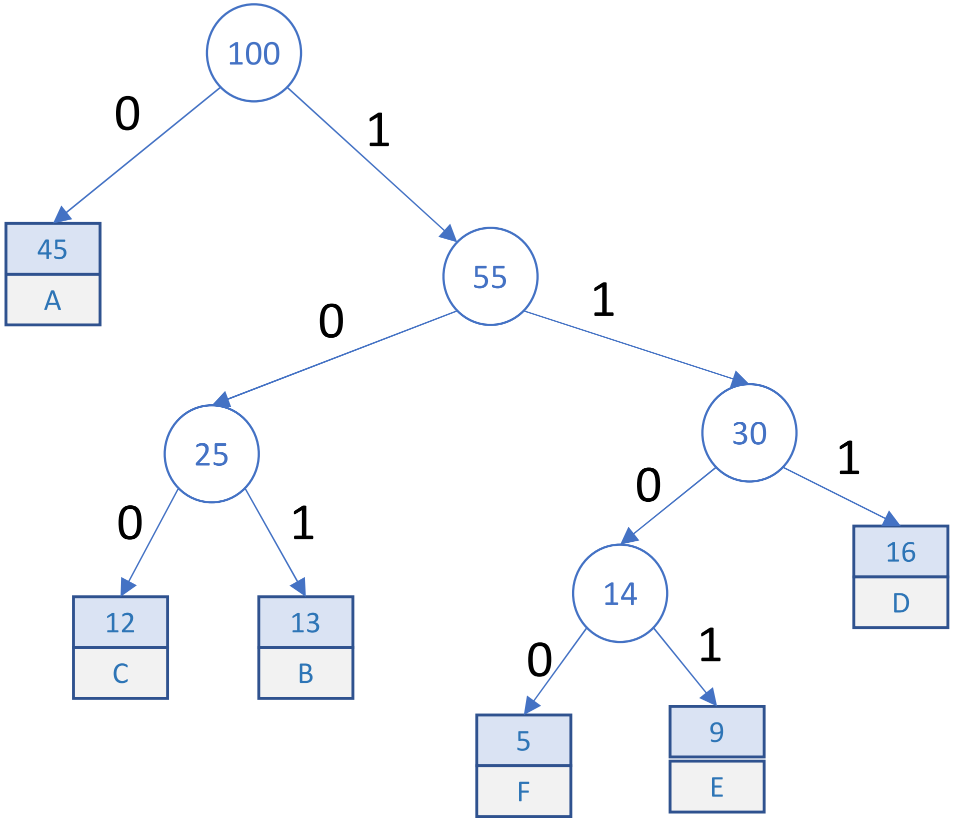

0 0 101 1101

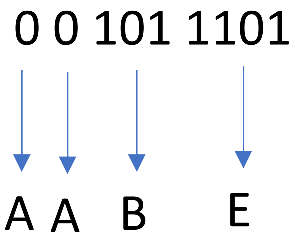
A A B

Huffman Coding

- Construct the Huffman Tree for the character set given below:

Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5
Code	0	101	100	111	1101	1100

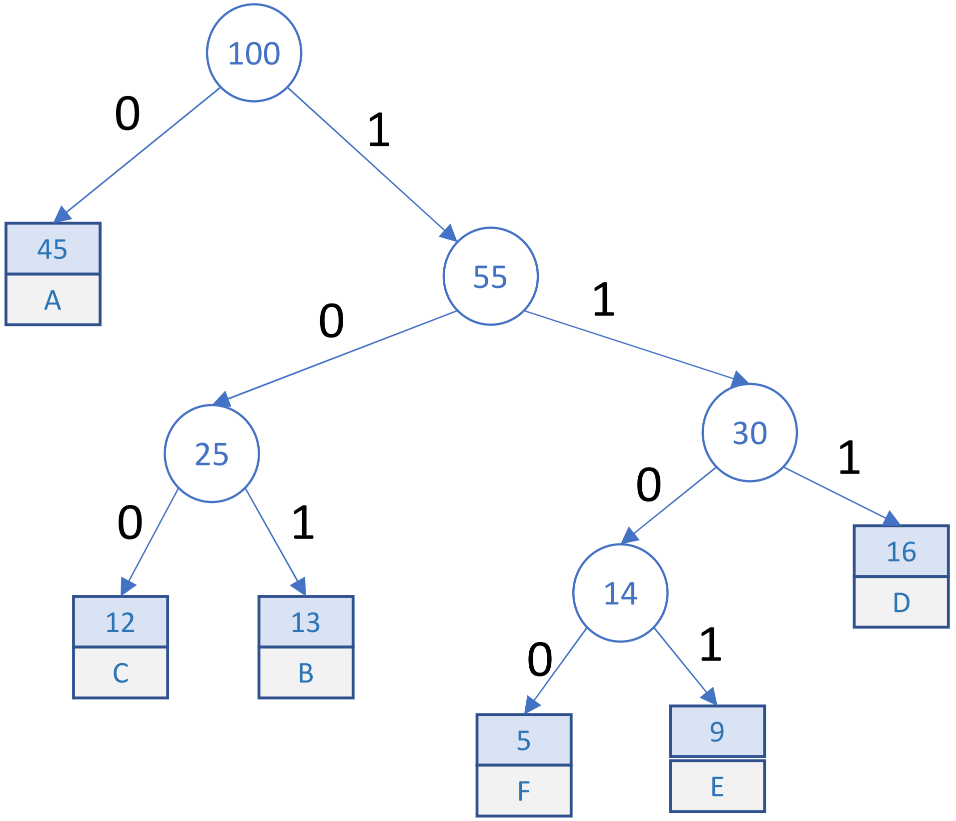




Huffman Coding

- Construct the Huffman Tree for the character set given below:

Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5
Code	0	101	100	111	1101	1100

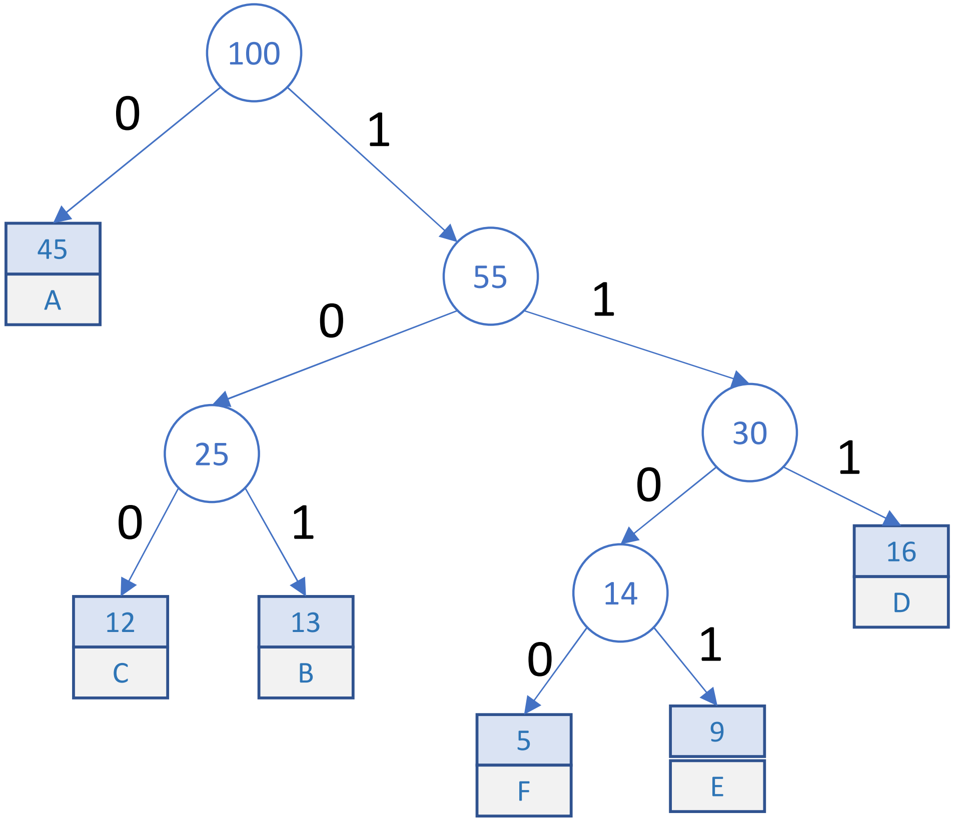


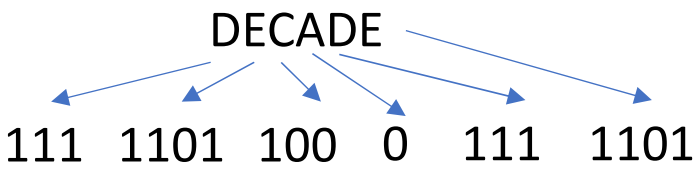
Encode DECADE

Huffman Coding

- Construct the Huffman Tree for the character set given below:

Letter	A	B	C	D	E	F
Frequency	45	13	12	16	9	5
Code	0	101	100	111	1101	1100


100
0
1
45
A
55
0
1
25
30
0
1
0
1
12
C
13
B
14
16
D
0
1
5
F
9
E


DECADE
111 1101 100 0 111 1101