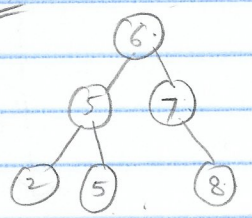
MIDTERM
CSC 6580
PRABHJOT KAUR

Question 1: (10) Write a modified version of Inorder traversal for a BST that will print the keys in reversed order.

Solution:

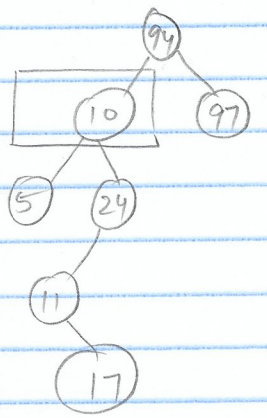
```
INORDER-TREE-WALK (x)
    INORDER-TREE-WALK (x.right)
    print x.key
    INORDER-TREE-WALK (x.left)
```

Ex



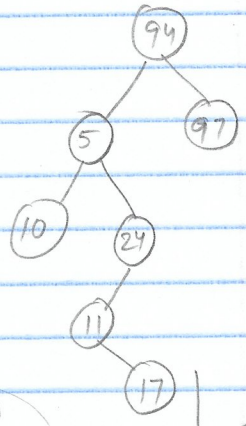
8 7 6 5 5 2

(11) from the following BST, delete the node with key 10.

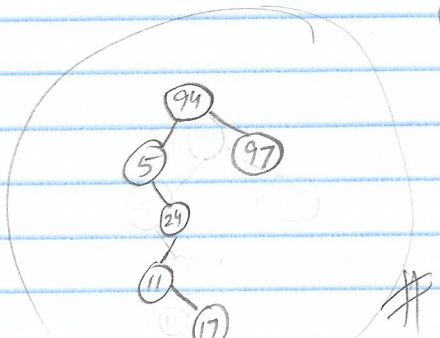


Case 2

Swap (10) with max [10.(left)] = 5

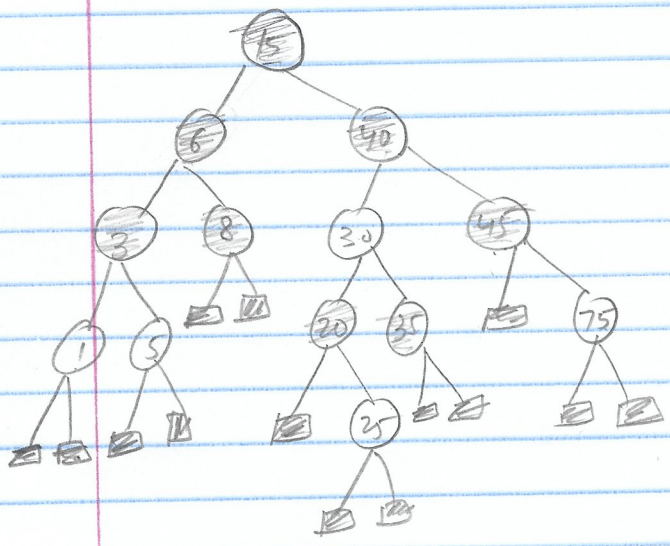


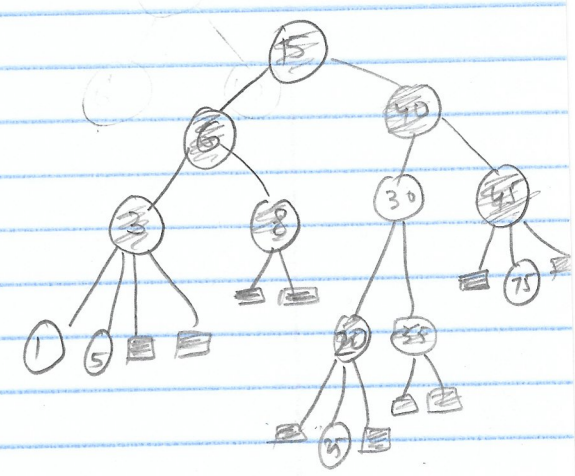
Case 0
10 has no children

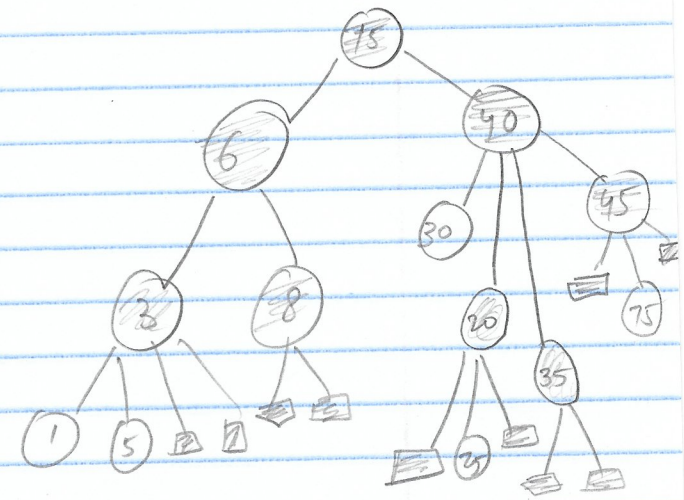


Question 2

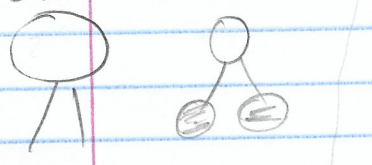
(i) Convert Red Black tree to a 2-3-4 tree

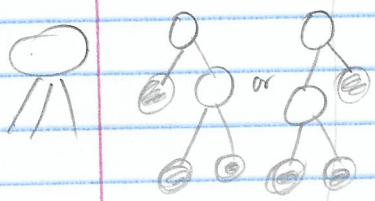


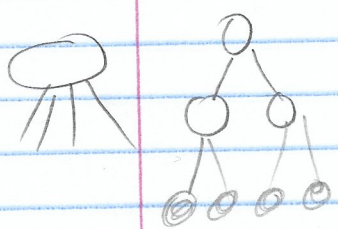




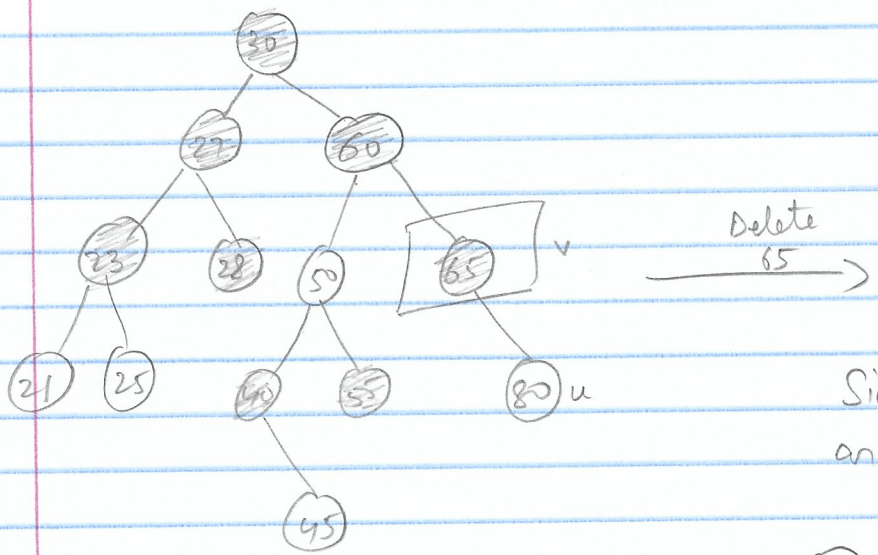
2-3-4 Red Black





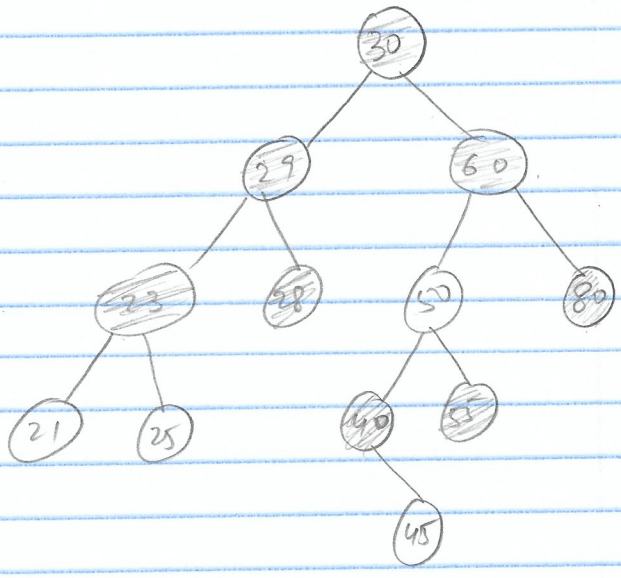


(ii) Delete 65



Delete 65 →

Simply, delete (65) and color (80) as black.



Question 3

(i) using optimal substructure property of LCS, write recursive form of the length of LCS of two sequences of characters.

Solution

let the length of LCS be $c[i, j]$ then the recursive form is defined below.

$$c[i, j] = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ c[i-1, j-1] + 1 & \text{if } i,j>0 \text{ and } x_i = y_j \\ \max(c[i, j-1], c[i-1, j]) & \text{if } i,j>0 \text{ and } x_i \neq y_j \end{cases}$$

The first case corresponds to when either X or Y sequences are empty.
The second case corresponds to when we have found a match between X & Y, then we increment the length by 1.
The third case corresponds to when when we have not found a match.

when match is found, we need to find LCS of X_{m-1} and Y_{n-1}.
when match is not found, we need to solve two subproblems:

(a) LCS of X_{m-1} and Y
(b) LCS of X and Y_{n-1}

whichever of these is longer is LCS of X and Y

(ii) Using DP construct the table to determine the length of LCS and to determine the LCS b/w the following two sequence of characters.

X = V P V P F Y P ⇒ m = 7

Y = P F V Y P Y ⇒ n = 6

Solution

Table c

			P	F	V	Y	P	Y
		y_j	1	2	3	4	5	6
	0	0	0	0	0	0	0	0
V	1	0	↑0	↑0	↖1	←1	←1	←1
P	2	0	↖1	←1	←1	←1	↖2	←2
V	3	0	↑1	←1	↖2	←2	←2	←2
P	4	0	↖1	←1	↑2	←2	↖3	←3
F	5	0	↑1	↖2	←2	←2	↑3	←3
Y	6	0	↑1	↑2	←2	↖3	←3	↖4
P	7	0	↖1	↑2	←2	↑3	↖4	↑4

(x_i down the rows, i; j across the columns)

length of LCS is 4

The LCS is P V P Y

Question 4 Consider the five symbol alphabet {A, B, C, D, –} with following occurrence frequencies in a text

symbol	A	B	C	D	–
frequency	0.35	0.1	0.2	0.2	0.15

(i) Construct Huffman coding tree

Solution

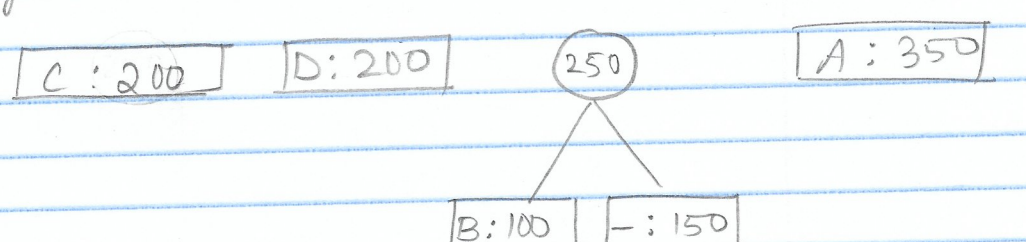
Arrange in increasing order of frequencies.

[B: 0.1] [–: 0.15] [C: 0.2] [D: 0.2] [A: 0.35]

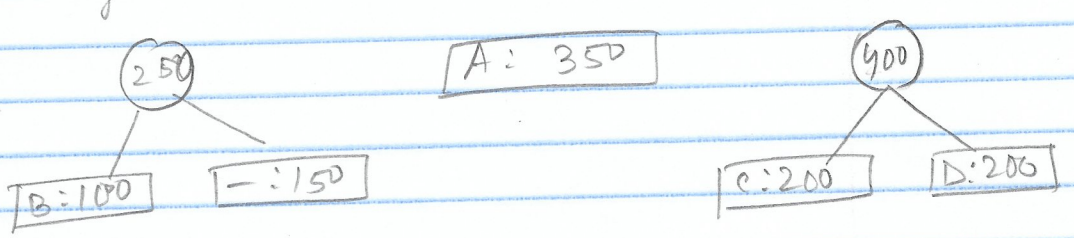
Multiply frequency by 1000 to ease calculation.

[B: 100] [–: 150] [C: 200] [D: 200] [A: 350]

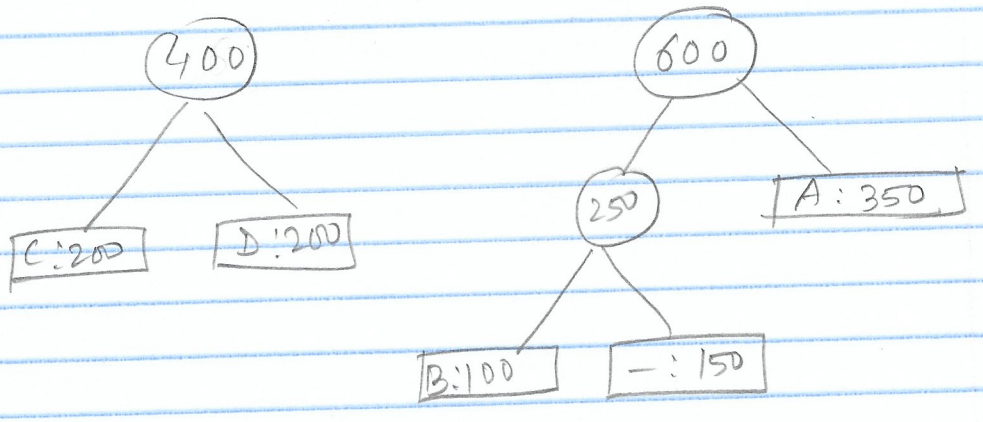
(a) Merge B and –



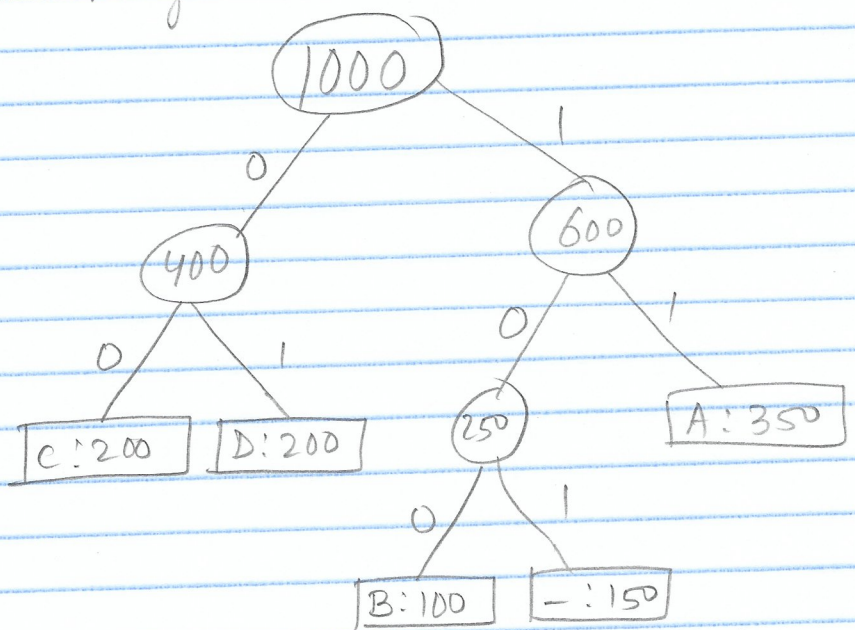
(b) Merge C and D



(c) Merge (250) and A



(d) Merge 400 and 600



This is the resulting Huffman coding tree.

(ii) The codewords

solution

B	=	100
—	=	101
C	=	00
D	=	01
A	=	11

(iii) The percentage size reduction to store

Solution:

For fixed length, we need at least 3 bits

fixed length file size:

$$3\left(\frac{0.35}{3} + 0.1 + 0.2 + 0.2 + 0.15\right) =$$

Variable length file size

$$= \underbrace{0.35 \times 2}_{A} + \underbrace{0.1 \times 3}_{B} + \underbrace{0.2 \times 2}_{C} + \underbrace{0.2 \times 2}_{D} + \underbrace{0.15 \times 3}$$

$$= 0.7 + 0.3 + 0.4 + 0.4 + 0.45$$

$$= 2.25$$

The percentage size reduction is

$$\left(\frac{3 - 2.25}{3}\right) \times 100 = \frac{0.75}{3} \times 100 = 25\%$$