ST-2 Solution

Subject Name - Data Structures

Subject code - RCS-303

Branch - CSE/IT

Section - CSE-1,2,3 | IT-1,2

Semester - 3

Session - 2017-2018 (ODD)

Faculty Name - Dr. Sachin, Ms. Swimpy Pahuja,
Ms. Shikha Jain, Ms. Swati Goel

Signature - Sjain..
(SHIKHA JAIN)

Reviewed
Vidushi

Q1 What is dequeue? Explain.

Ans Dequeue is doubly ended queue. In this queue we can make use of both the ends for insertion of the elements as well as both the ends for deletion of the elements. It is also circular in nature. We will implement dequeue using a circular array.

Q2. What is tail recursion?

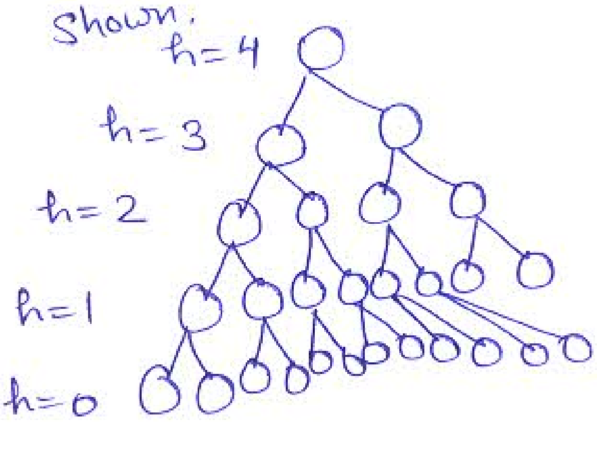
Ans. A function call is said to be tail recursive if there is nothing to do after the function returns except return its value.

a) It helps in improving the space & time efficiency of recursive function.

b) It optimizes the task of a compiler for executing the recursive function.

Q3 If there are 27 nodes in a complete binary tree, what will be its height and how many nodes will be in the last level?

Ans For 27 nodes the complete binary tree is as shown.



Height = 4

No. of nodes at last level = 12

Q4 If the Tower of Hanoi is operated n=11 disks, calculate the total number of moves.

Ans Total no of moves with n disks in Tower of Hanoi problem is $2^n - 1$

So for n=11 No of moves = $2^{11} - 1$

= 2048-1

= 2047

Q5 Define priority queue. How is it implemented?

Ans. Priority queue is a queue in which insertion and deletion is based on the priority of the element. Every node in the queue is associated with 3 fields.

```
struct element
{
int data;
int priority;
struct element *nxet;
};
```

It is implemented using the following 2 methods-

a) Through Queue

b) Through Sorted array

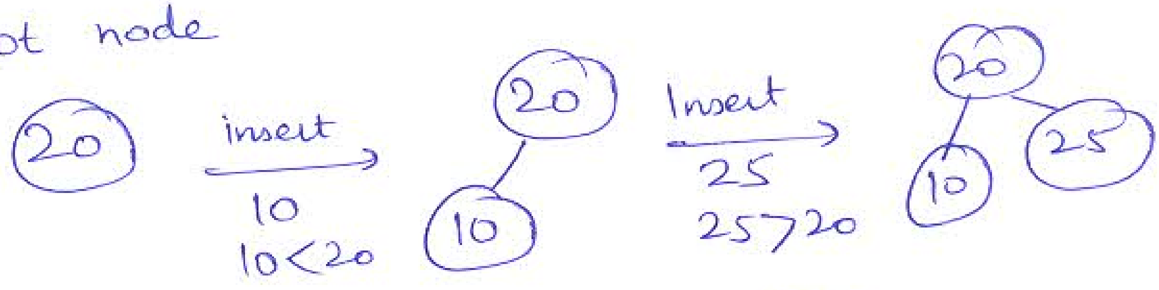
Section B

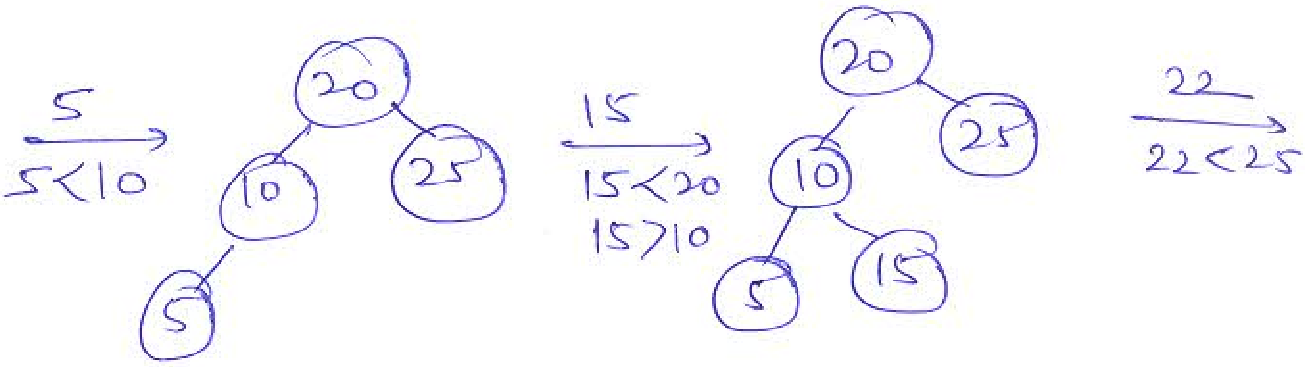
Q6 Create BST for the following data, show all steps

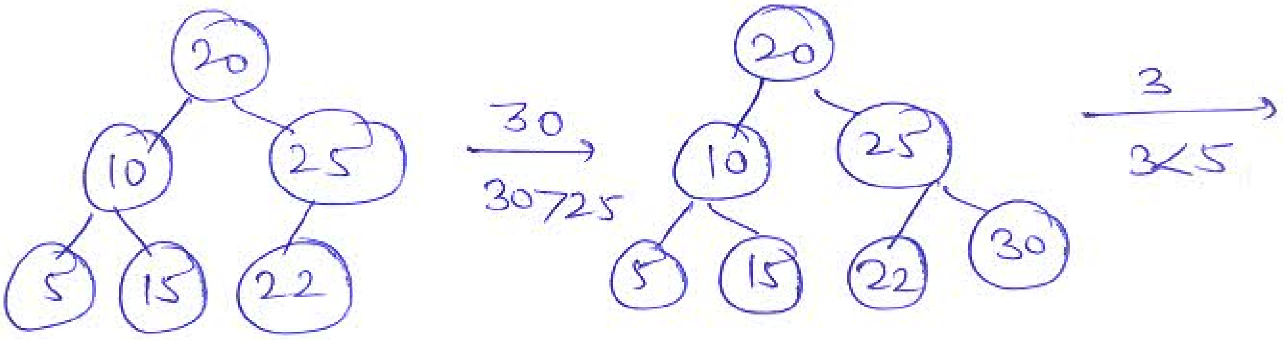
Insert: 20, 10, 25, 5, 15, 22, 30, 3, 14, 13

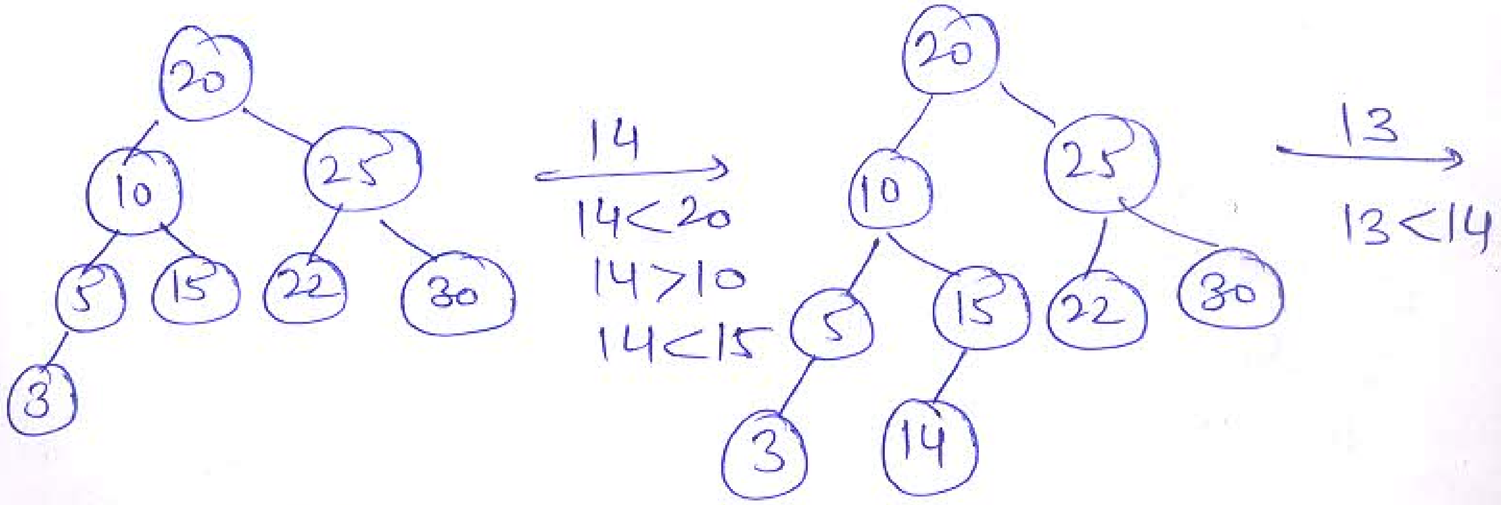
Delete: 20, 22, 10

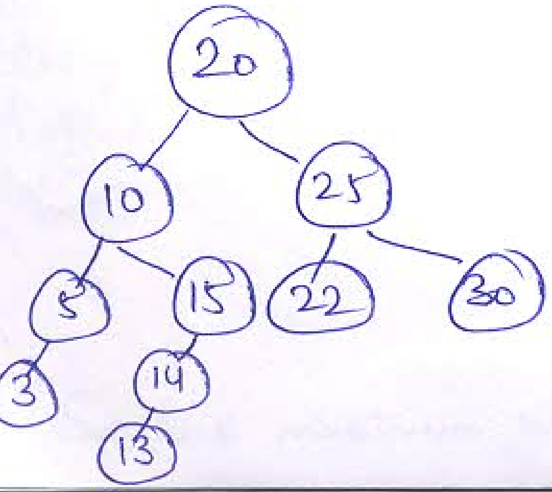
Ans. Insert 20 so 20 node will be the root node







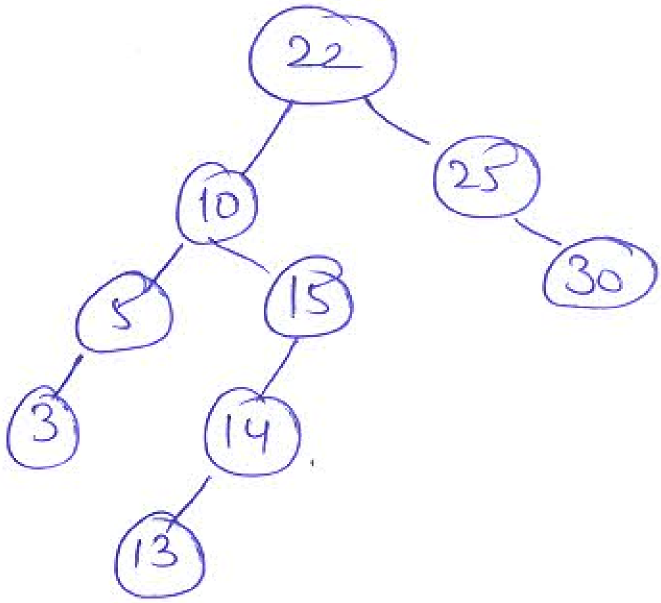




Delete 20 →

2 child Case-C

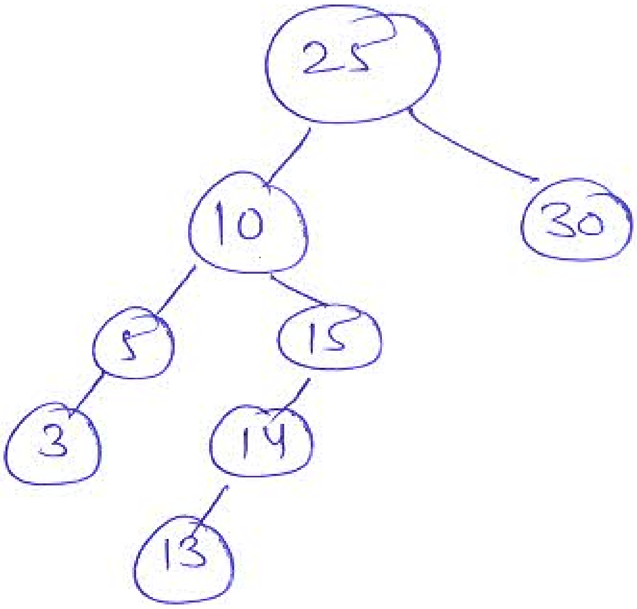
Inorder succ of 20 is 22



delete 22 →

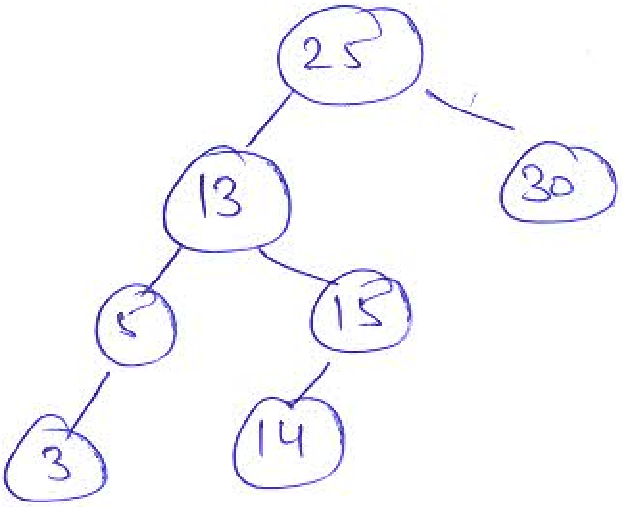
Case-C

Inorder succ of 22 is 25



delete 10 →

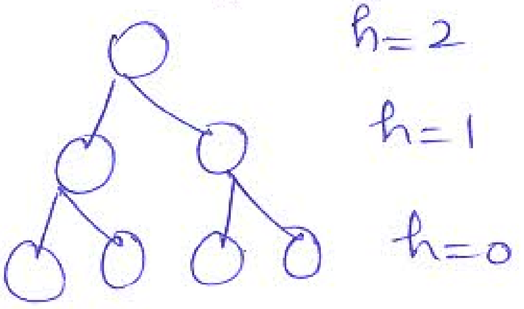
Inorder succ of 10 is 13



Ans.

Q7 Prove that the maximum number of nodes in a binary tree of height h is $2^{h+1}-1$.

Ans



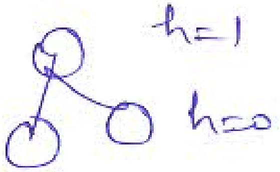
Height of tree = 2

maximum no of nodes = 7

$2^{2+1}-1$ is true

We use Method of Induction

for h = 1



no of node = $2^{2}-1$
= 3 true

for h=2

no of nodes = $2^{2}-1$ true
= 7

Let suppose this hypothesis is true for some $k<h$

for h=K the maximum no of nodes = $2^{K+1}-1$

Induction Step

No of nodes in T = $T_L + T_R + 1 \Leftarrow$ root node

$= 2^{h_L+1} - 1 + 2^{h_R+1} \cancel{-1} + \cancel{1}$

∴ Height of subtree is less than total height of the tree

$= 2^{h_L+1} + 2^{h_R+1} - 1$

$= (2(2^{\max(h_L+1,\, h_R+1)}) - 1)$

$= 2\,2^{h(T)} - 1$

$= 2^{h(T)+1} - 1$

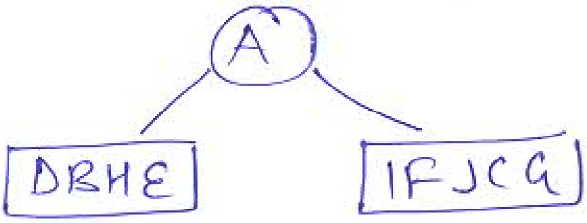
So the maximum no of nodes of a binary tree with height h = $2^{h+1}-1$

Q8. Generate a binary tree for the following traversal sequences

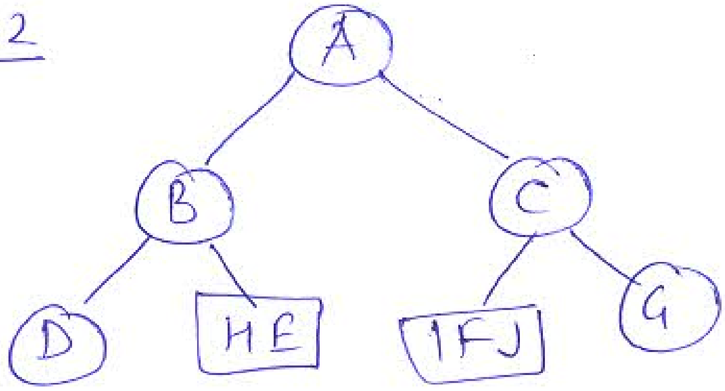
INORDER D B H E A I F J C G

PREORDER A B D E H C F I J G

Ans. The root node is A according to Preorder traversal

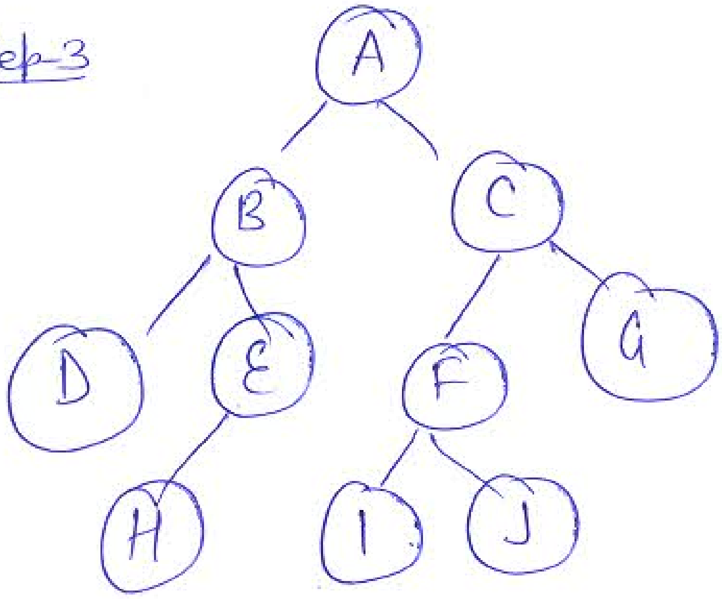


Step-2



B is the root among DBHE acc. to preorder traversal.
Slly C is the root.

Step-3



Final tree according to the INORDER and PREORDER

Q9. Write a C program to implement the array representation of circular queue.

Ans
```
# include<stdio.h>
# include <conio.h>
# define max 4
void insert (int val);
int del();
void display ();
int que[max] , front=-1, rear=-1;
void main()
{
int c, val;
do
{
printf(" Enter 1) to insert 2) delete 3) display 4)");
scanf("%d", &c);
switch(c)
{
Case1: printf("input value");
       scanf("%d", &val);
       insert (val);
         break;
Case 2:    val= del();
       if ( val != -1)
printf(" value deleted is %d", val);
       break;
```

```
     case 3:  display();
              break;
   default:   exit(0);
     }}while(1);
       }

  void insert (int val)
   {
 if ( front == (rear+1)% size)
       {
 printf(" queue is full: overflow");
      return;
    }
 if  (front==-1)
     {
    front=0;
    rear=0;
     }
   else
   rear= (rear+1) % size;
    que[rear]= val;
      }
   int del()
    {
    int val;
  if (front==-1)
     {
 printf(" queue is empty");
```

```
        return -1;
          }
      val= que[front];
     if (front== rear)
        {
     front = -1;
      rear= -1;
   else
    front= (front+1)% size;
   return val;
      }
  void display()
      {
 printf(" elements of queue are");
 if (front <= rear)
     {
for(i=front; i<= rear; i++)
printf("%d", que[i]);
    }
else
   {
 for (i= front; i<= size -1; i++)
 printf("%d", que[i]);
 for (i=0; i<= rear; i++)
  printf ("%d", que[i]);
     }
    }
```

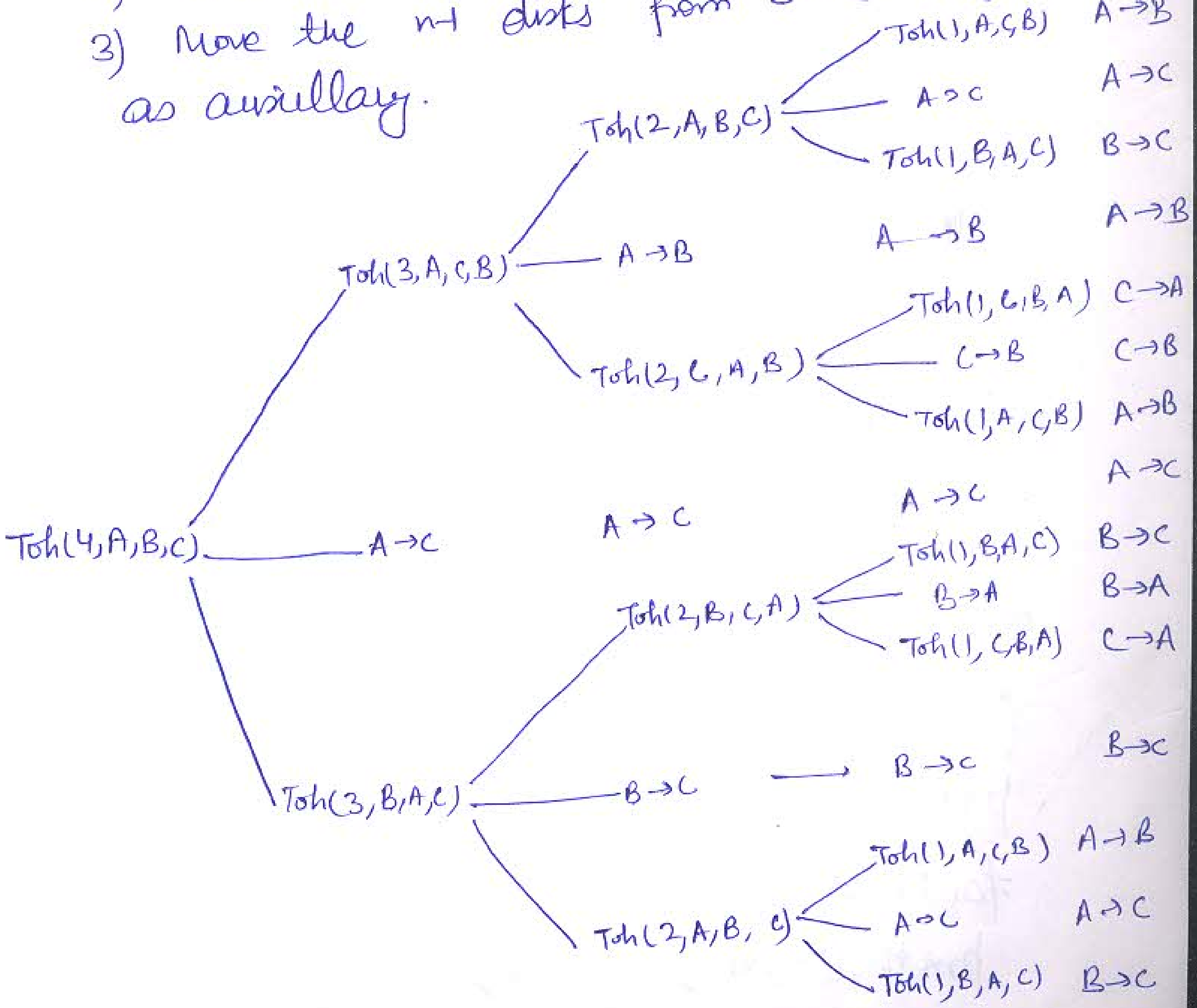
Q10 Write an algorithm for finding solution to the Tower of Hanoi problem. Explain the working of your algorithm with 4 disks with diagram.

Ans. Algo

1) Move top n-1 disks from A to B using C as auxillary.

2) Move the remaining disks from A to C

3) Move the n-1 disks from B to C using A as auxillary.



Q11 A) Convert this infix expression into prefix expression ⑤

$a-b+c*(d/e-(f+g))$

B) Evaluate the postfix expression

$32 \wedge 5 * 32 * 3 / 5 + -$

Ans A) Step 1 reverse $((g+f)-e/d)*c+b-a$

Symbol	Stack	postfix
(	(	—
(	((	—
g	((	g
+	((+	g
f	((+	gf
)	(	gf+
-	(-	gf+
e	(-	gf+e
/	(-/	gf+e
d	(-/	gf+ed
)	ϕ	gf+ed/-
*	*	gf+ed/-
c	*	gf+ed/-c
+	+	gf+ed/-c*
b	+	gf+ed/-c*b
-	+-	gf+ed/-c*b
a	+-	gf+ed/-c*b a
	ϕ	gf+ed/-c*b a-+

(∵ we don't pop same priority operators)

Required prefix expression +-a b*c-/de+fg

+-ab*c-/de+fg

B 3 2 ^ 5 * 3 2 * 3/5 + -

Ans

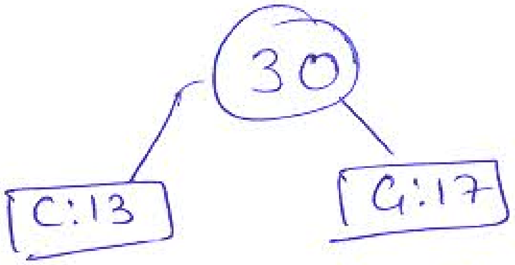
Input Symbol	Stack
3	3 push it onto stack
2	3, 2 ↑top
^	3^2 = 9 next to top OP
5	9, 5 top
*	9×5= 45
3	45,3 push
2	45,3, 2 ↑top push
*	45, 6 ↑top 3*2=6
3	45, 6, 3 next to TOP OPERATOR
/	45, 6/3 = 45, 2 TOP
5	45,2,5 5+2=7 push back onto stack
+	45, 7
—	45-7 = $\boxed{38}$ Ans

Q12 What is variable length encoding scheme. Draw a Huffman tree and encode each character for the following symbols whose frequency of occurrence is a message is stated as

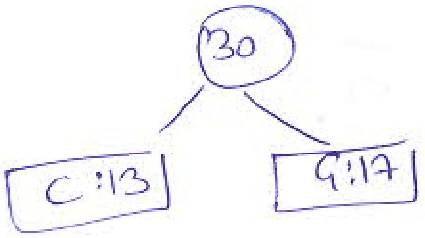
A	B	C	D	E	F	G	H	T	U
24	55	13	67	88	36	17	61	24	76

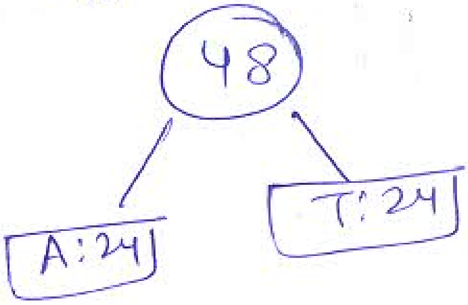
Ans Variable length Encoding scheme encode each character in a message with different length whereas fixed length encoding encode each character with same length

Choose 2 minimum number

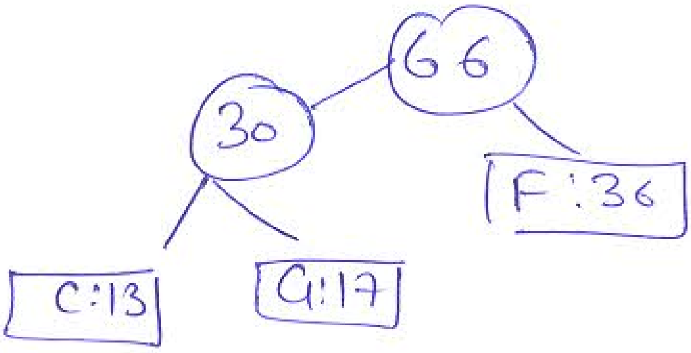


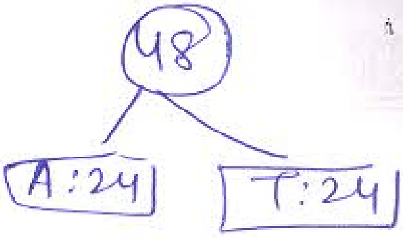
A	B	D	E	F	H	T	U	
24	55	67	88	36	61	24	76	30



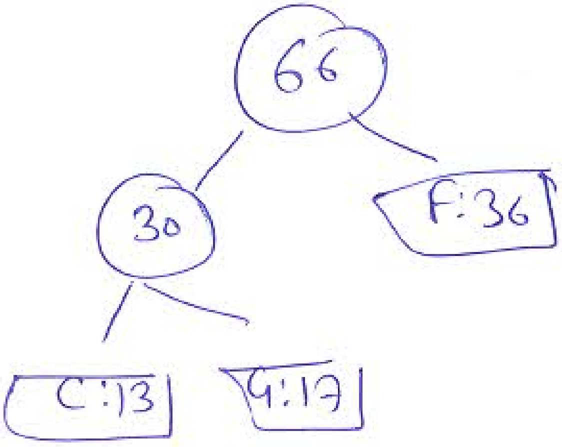


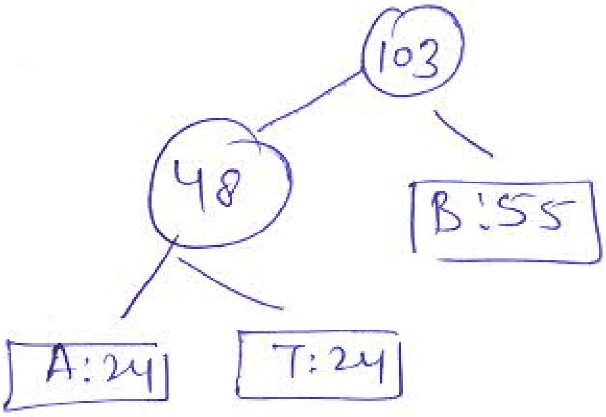
B	D	E	F	H	U		
55	67	88	36	61	76	30	48



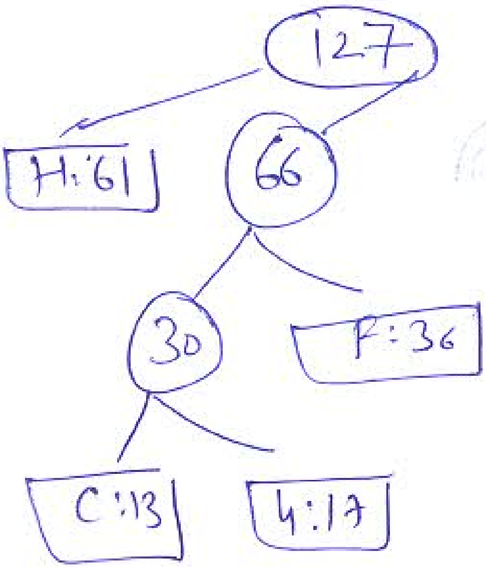


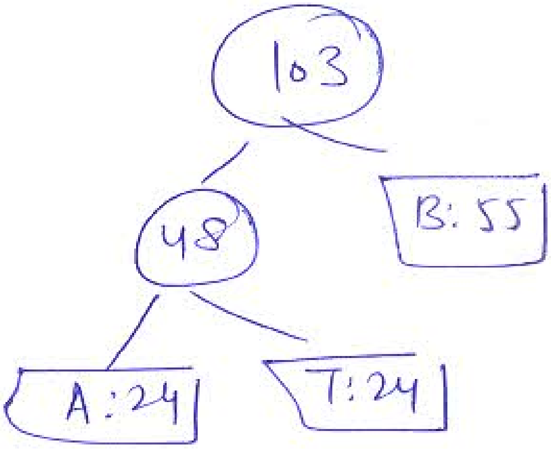
B	D	E	H	U		
55	67	88	61	76	48	66



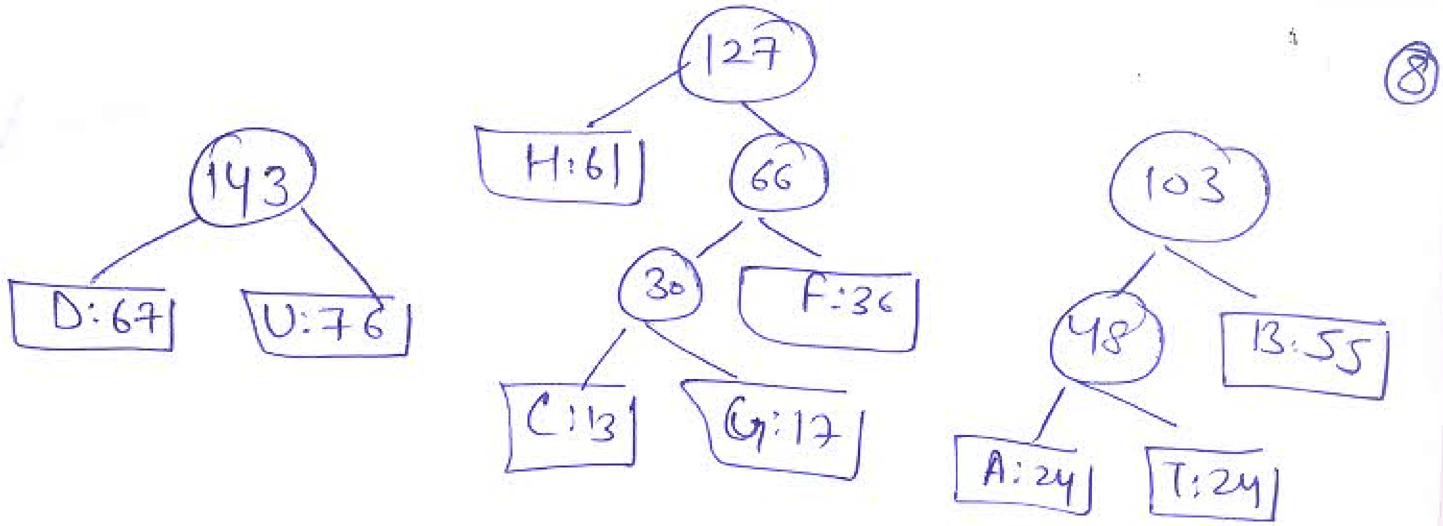


D	E	H	U		
67	88	61	76	66	103

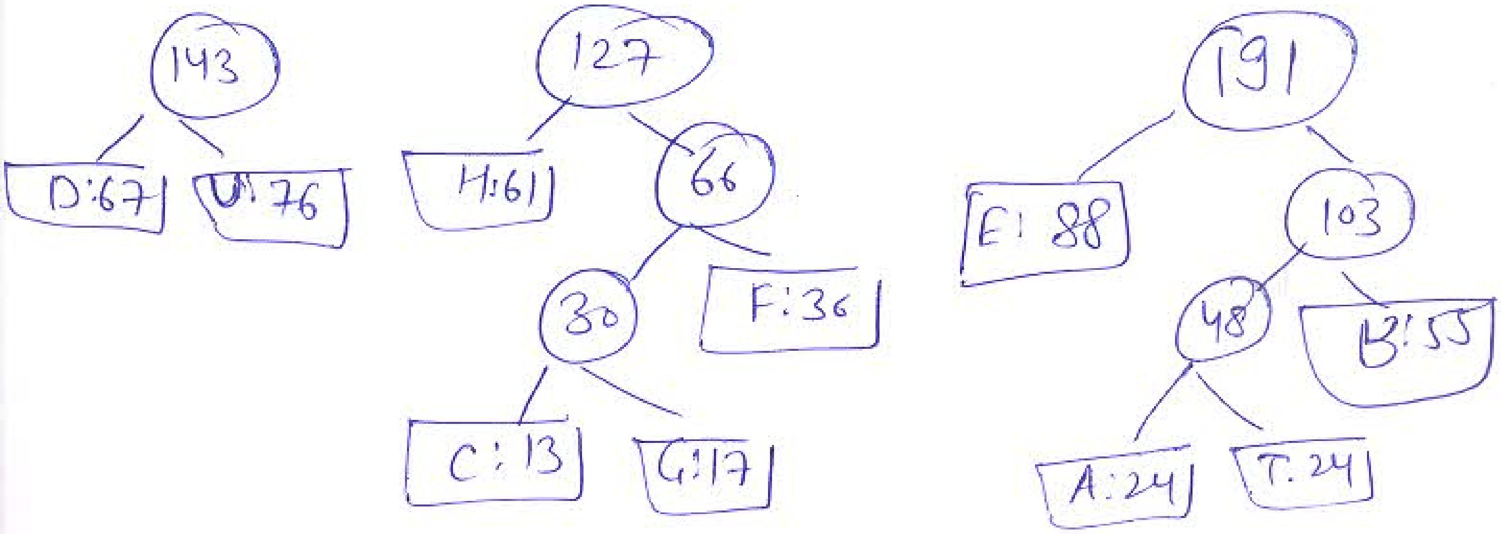




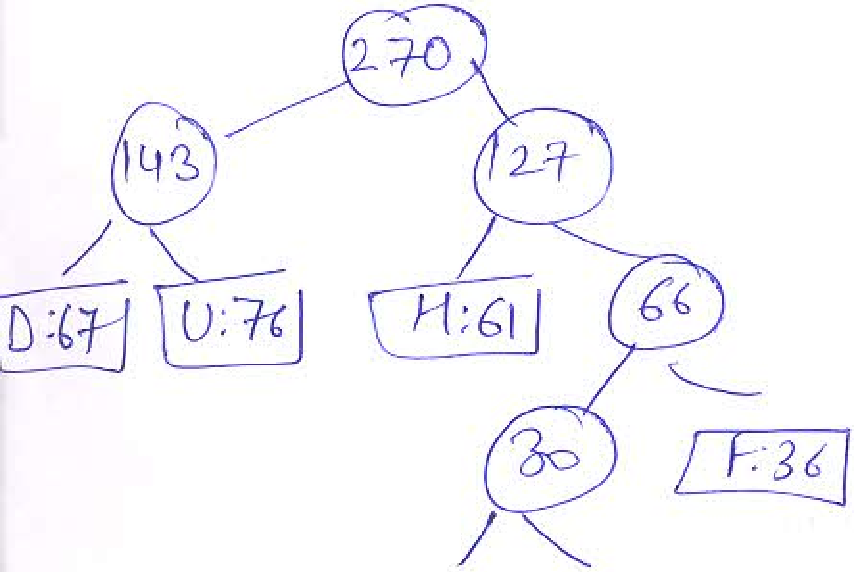
D	E	U		
67	88	76	103	127

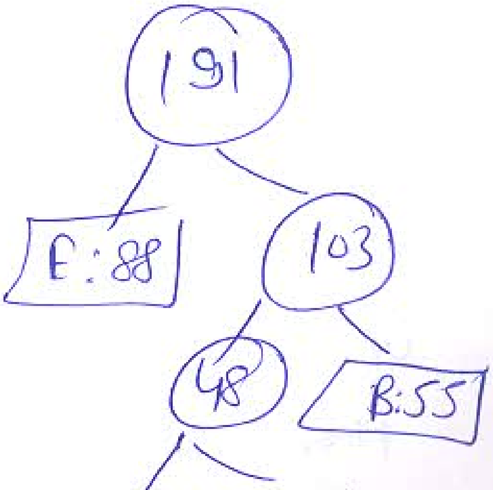


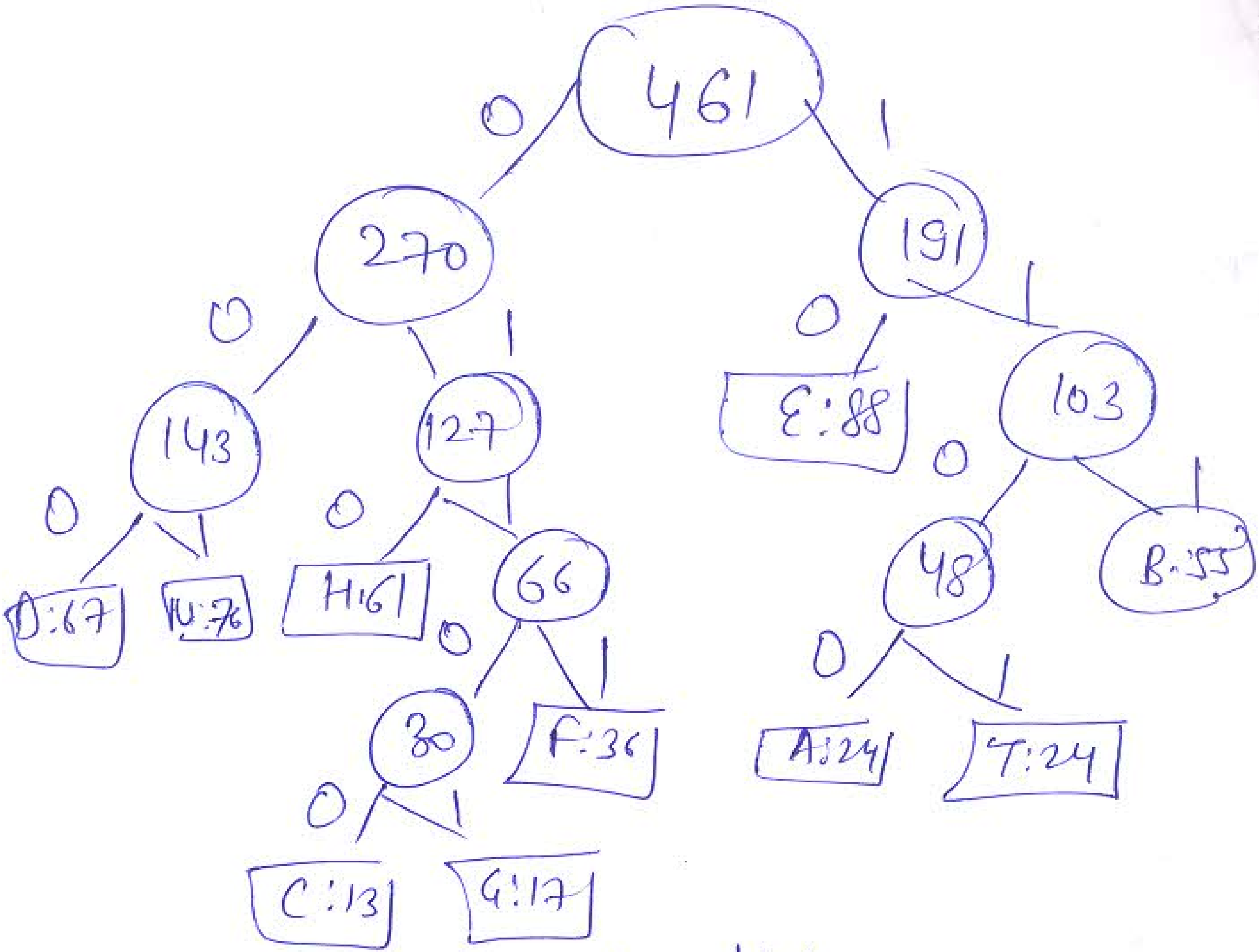
E
88 143 127 103



143 127 191







Encoding

A; 24	1100
B: 55	111
C:13	01100
D:67	000
E:88	10
F:36	0111
G:17	01101
H:61	010
T: 24	1101
U:76	001