STUDENT PORTFOLIO

Insert Photo

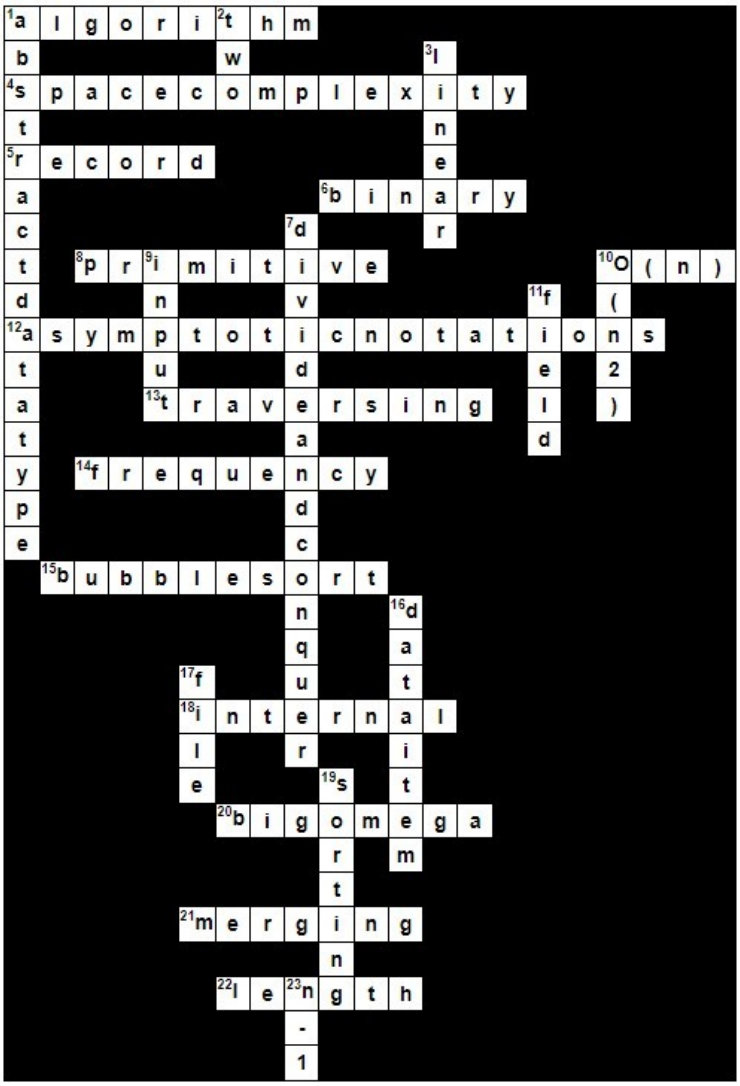
Name: ABHINAYA SREE TAKURU
Register Number: RA2111030010051
Mail ID: tt8059@srmist.edu.in
Department: B.Tech CSE
Specialization: CYBERSECURITY
Semester: III

Subject Title: 18CSC201J Data Structures and Algorithms

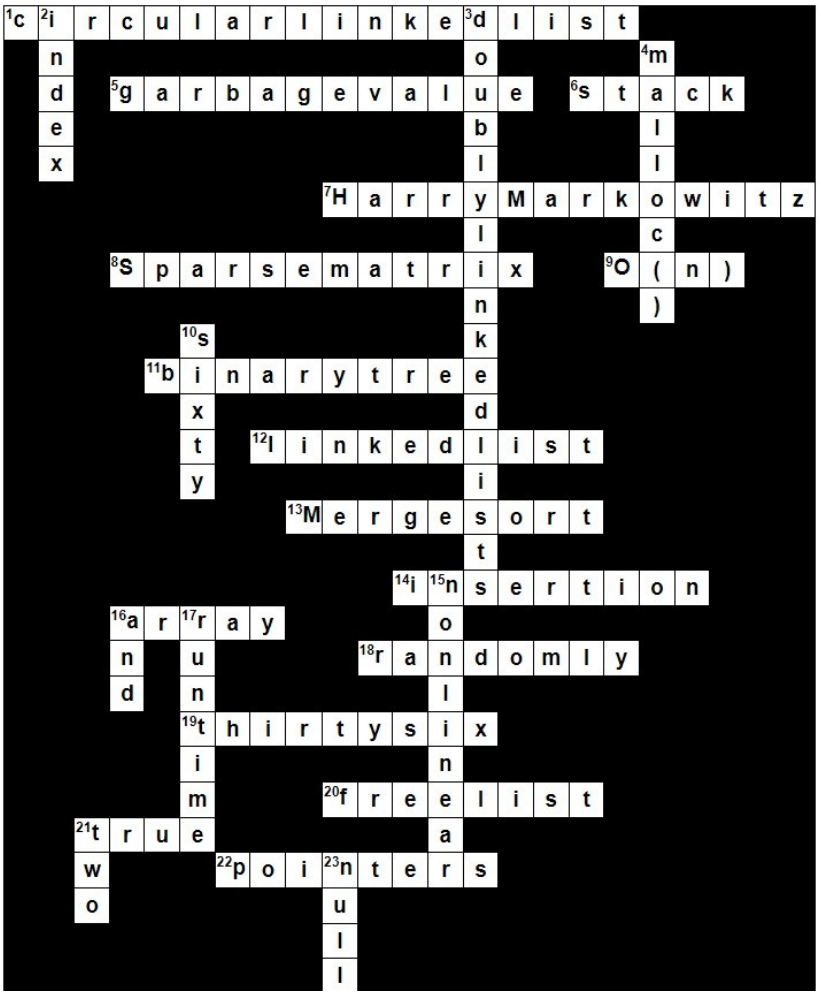
Handled By: Dr.M.Jeyaselvi

Assignment – CrossWord Puzzle (Unit 1,2,3, & 4)
(Write about the assignment questions and how u solved differently)

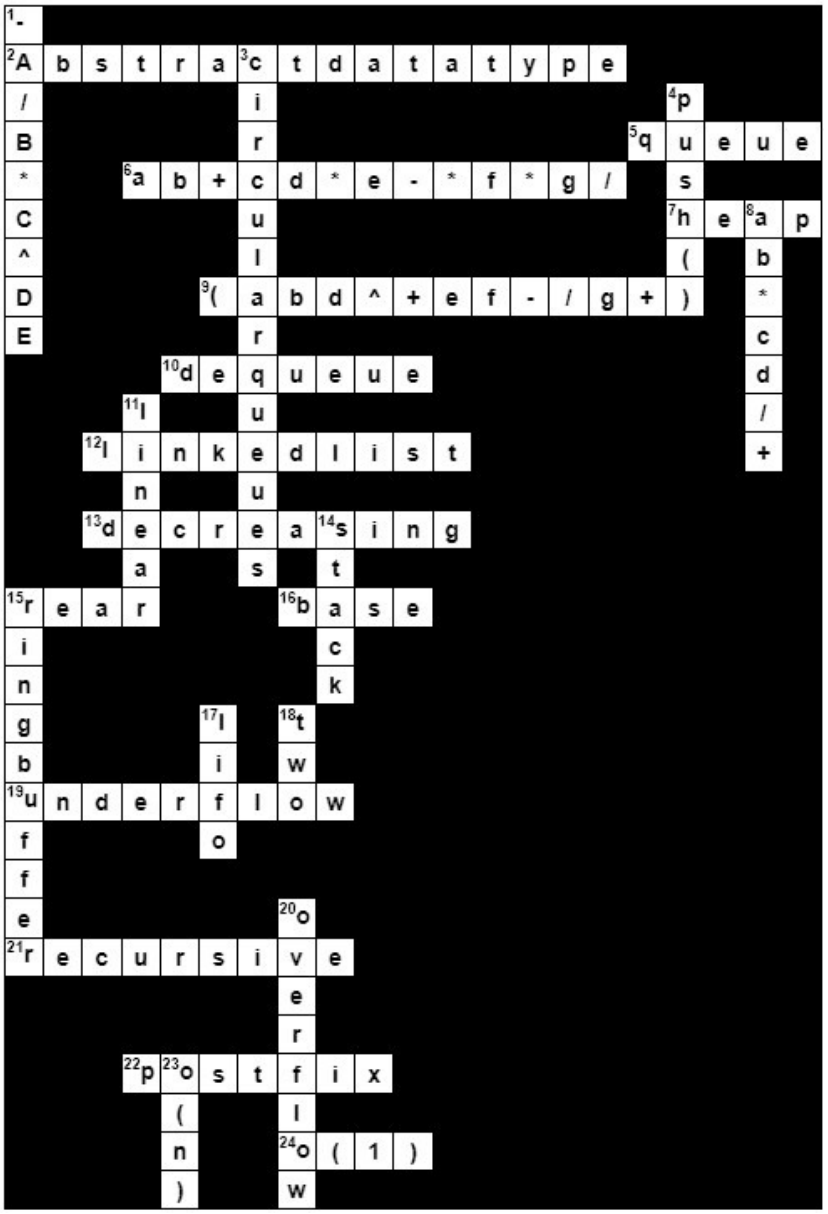
UNIT-1



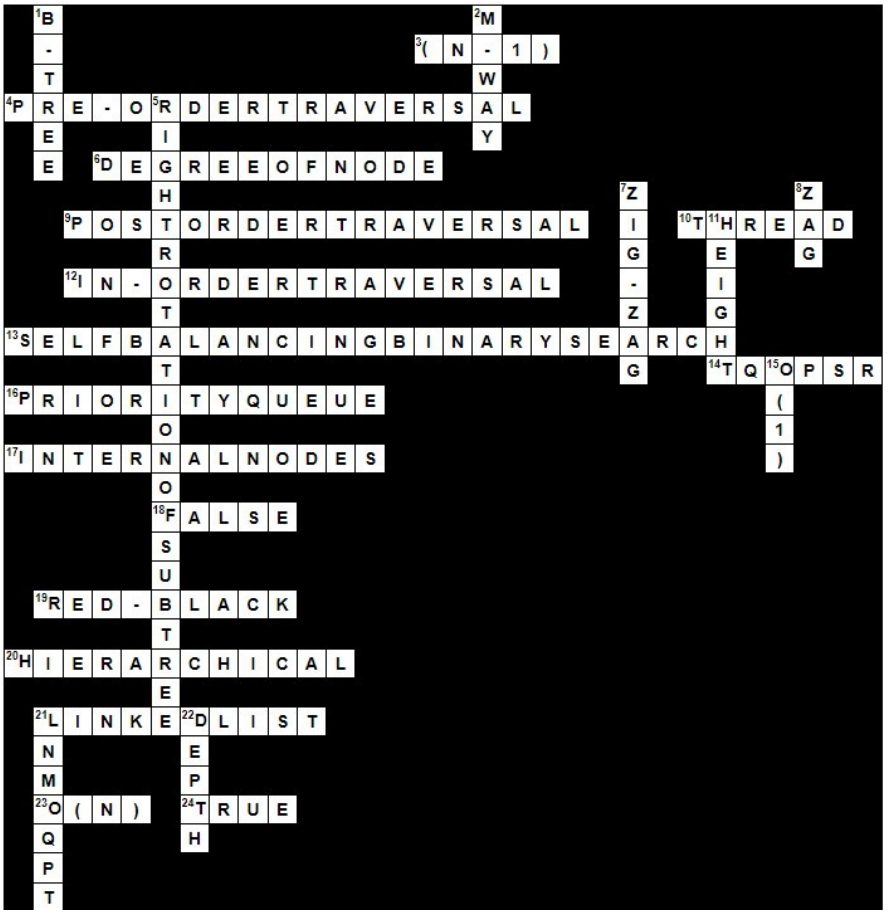
UNIT-2


1 c 2 i r c u l a r l i n k e d 3 l i s t
n o 4 m
d 5 g a r b a g e v a l u e 6 s t a c k
e b l
x l l
7 H a r r y M a r k o w i t z
l c
8 S p a r s e m a t r i x 9 O (n)
n)
10 s k
11 b i n a r y t r e e
x d
t 12 l i n k e d l i s t
y i
13 M e r g e s o r t
t
14 i 15 n s e r t i o n
16 a r 17 r a y o
n u 18 r a n d o m l y
d n l
19 t h i r t y s i x
i n
m 20 f r e e l i s t
21 t r u e a
w 22 p o i 23 n t e r s
o u
l
l

UNIT-3


1-
2Abstract3ctdatatype
/ i 4p
B r 5queue
* 6ab+cd*e-*f*g/ s
C u 7he 8ap
^ l (b
D 9(abd^+ef-/g+) *
E r c
10deque d
11l u /
12linkedlist +
n u
13decrea14sing
a s t
15rear 16base
i c
n k
g 17l 18t
b i w
19underflow
f o
f
e 20o
21recursive
e
r
22p 23ost fix
(l
n 24o(1)
) w

UNIT-4



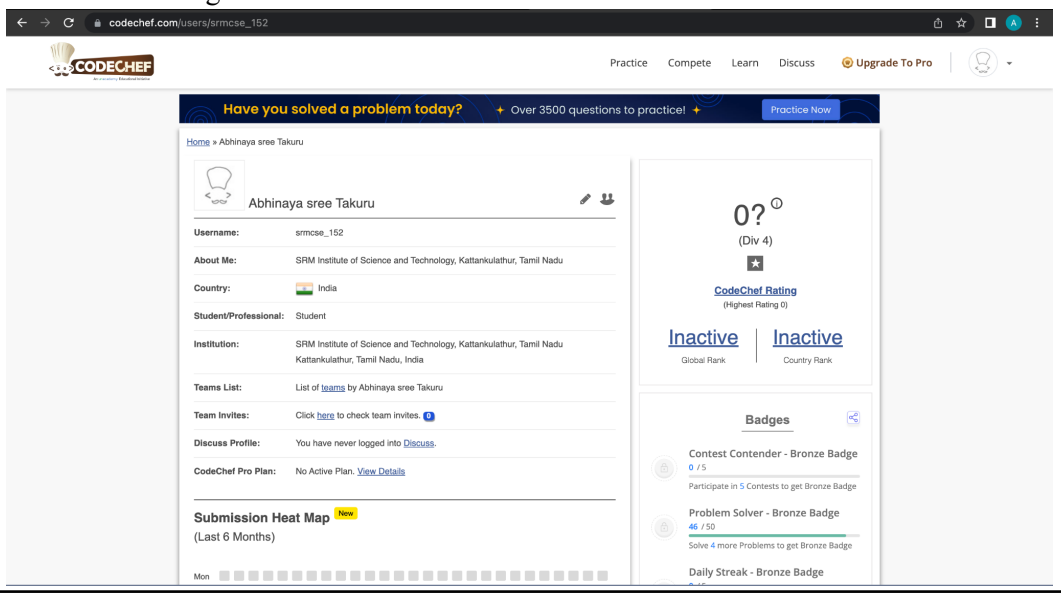
Assignment

(what is the most interesting part in the assignment)

It was really informative and it was a kind of a recap or summary for what all we were taught
The crossword puzzle was really helpful in order to recap all the topics as it covered mostly all the topics which we were taught.

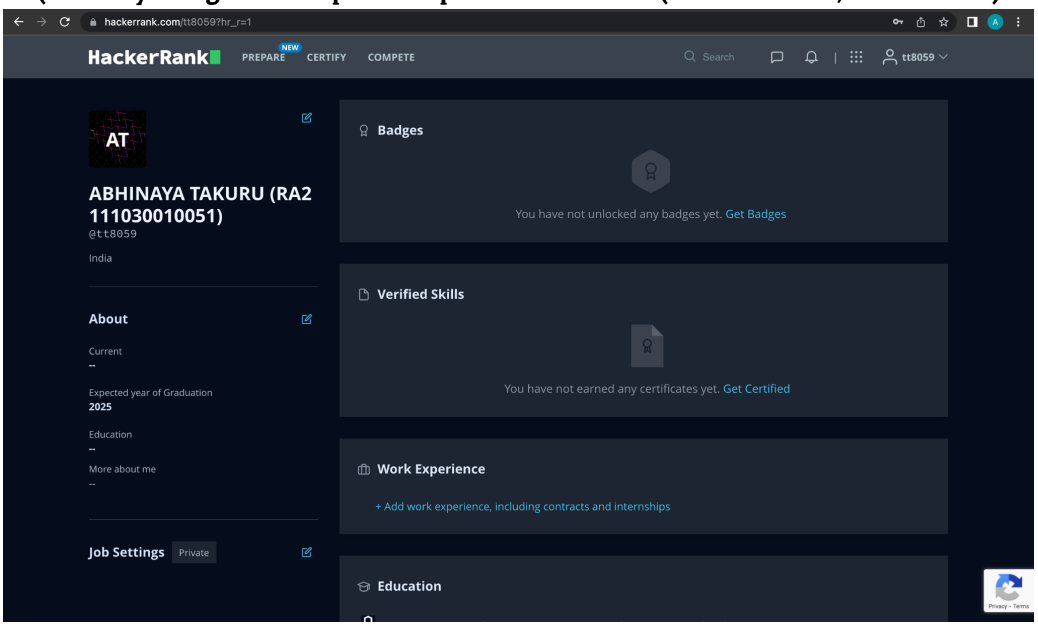
Codechef Achievements

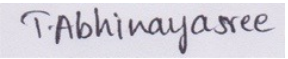
Completed all the weekly assignments and practice problems given by the faculty of Data Structures and Algorithms.



Any other

(Write if you registered or practise apart from Codechef(ex. Hackerrank, Leetcode etc.)





Signature

Note: Enclose the assignment and relevant certificates along with the profile

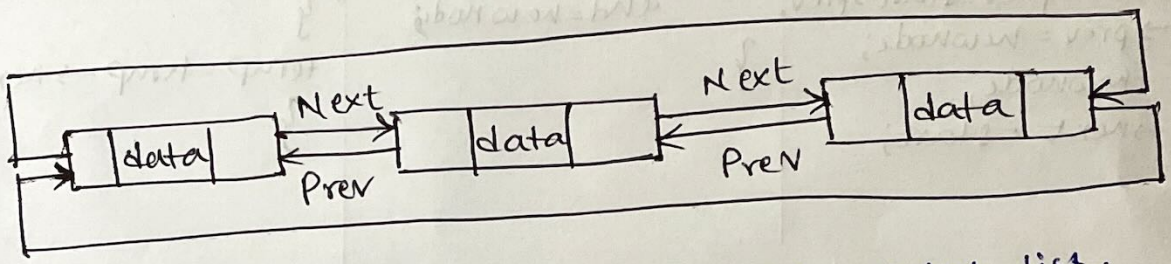
RA2111030010051
T-ABHINAYA SREE

DSA ASSIGNMENT-1

1. Definition of Circular Doubly Linked List.

A doubly circular linked list is a linked list that has the features of both the lists i.e Doubly linked list and circular linked list. In a doubly circular linked list, two nodes are connected or linked together by the previous & the next pointers, and the last node of doubly circular linked list is connected to first node of doubly circular linked list.

2. Graphical Representation of circular Doubly Linked List:



3. Algorithm or Pseudo code of Circular Doubly linked list.

Algorithm:

i) Create a structure of node of doubly circular linked list.
ii) To create a node first initialize a new node with the help of malloc function.
iii) If START == NULL, then START = newnode & newnode = end. Else, end → next = newnode & newnode → Prev = end.
iv) Set newnode → next = start; start → prev = newnode

Traversal:

i) Set temp = start
ii) temp → next ≠ start, Print temp → data.

Insertion:

i) Initialize a newnode
ii) If start == NULL, set start = newnode & create link
iii) Else newnode → next = start; start → prev = newnode; newnode → prev = end; start = newnode.

Deleting:

i) Specify temp points to start.
ii) Then start = start → next
iii) Then setfree(temp)
iv) start → prev = end

4. CODE FOR INSERTION AND DELETION:

Insert front:

```
void infront(){
initialize();
if(start==NULL){
start=newnode;
newnode->next=start;
newnode->prev=start;}
else{
newnode->next=start;
newnode->prev=start->prev;
start->prev=newnode;
start=newnode;
end->next=start;
}
```

Insert last:

```
void inlast(){
initialize();
end=start;
while(end->next!=start)
{end=end->next;}
end->next=newnode;
newnode->prev=end;
newnode->next=start;
start->prev=newnode;
end=newnode;
}
```

Insert middle:

```
void middle(){
int x;
scanf("%d",&x);
temp=start;
while(temp!=null){
if(temp->data==x){
initialize();
temp->next=newnode;
}
temp=temp->next;
}
```

Deleting front

```
void delfront(){
temp=start
start=start->next;
free(temp);
start->prev=end;
end->next=start;}
```

Deleting last

```
void dellast(){
end=temp;
end=end->prev;
free(temp);
end->next=start;
start->prev=end;
```

Deleting Middle

```
void del_mid(){
int x; scanf("%d",&x);
temp=start;
while(temp!=null){
if(temp->data==x){
temp->prev->next=temp->next
temp->next->prev=temp->prev
free(temp);}
temp=temp->next;}
```

ADVANTAGES OF CIRCULAR DOUBLY LINKED LIST:

- List can be traversed from both directions.
- Each of data manipulation.
- Jumping from start to end or vice versa — O(1) time.

DISADVANTAGE OF CIRCULAR DOUBLY LINKED LIST:

- Requires additional memory.
- More complex than highly linked list.
- If not used properly, then problem of infinite loop can occur.

RA2111030010051

T. ABHINAYA SREE

DSA ASSIGNMENT - 2

1. Definition of binary search tree.

- Binary Search Tree is a node-based binary tree data structure which has the following properties:
  - The left subtree of a node contains only nodes with keys less than the node's key.
  - The right subtree of a node contains only nodes with keys greater than the node's key.
  - The left & right subtree each must also be a binary search tree.

2. Algorithm or Pseudo code of Binary Search Tree.

i) If tree is empty (no root), create a node holding key K as root.

ii) Set curr_node = root_node

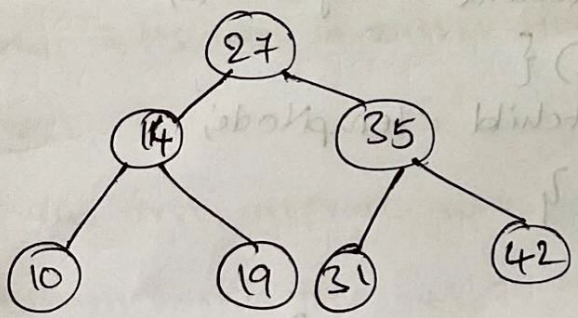
iii) If K == Curr_node key, done
No duplicate keys in a BST

iv) If K < curr_node key
key must go in left subtree.
If curr_node == NULL, create a node holding K as left child of curr_node. Else, set curr_node = curr_node left.

v) else if curr_node.right == NULL, create a node holding K as left child of curr_node else set curr_node = curr_node right.

vi) Go to (iii).

3. Graphical Representation of BST.



4. Code for INSERTION & DELETION.

INSERTION

```
void insert(int data) {
struct node* tempNode =(struct node*) malloc size of(struct Node);
struct node* current;
struct node * parent;
tempNode -> data = data;
tempNode -> leftchild = NULL;
tempNode -> rightchild = NULL;
if (root ==NULL) {
    root = tempNode; }
else{
     current = root;
     parent = NULL;
 while(1) {
     parent= current;
if(current ==NULL) {
   parent -> leftchild = tempNode;
   return;
}}
 else { current=current -> rightchild;
if(current ==NULL) {
   parent -> rightchild =tempNode;
return; }}}}}
```

DELETION

```
void deletion(Node *&root, int item)
{ Node * parent= NULL;
  Node * curr = root;
 search (curr, item, parent);
if (curr ==NULL)
return;
```

```
if (curr->left==NULL && curr->right==NULL)
{if(curr!=root){if(parent->left==curr)
parent->left=NULL;
else {parent->right=NULL;}}
else if(curr->left && curr->right){
  Node* succ=findminimum(curr->right);
   int var=succ->data;
deletion(root,succ->data);
   curr->data=var;}
else{ Node*child=(curr->left)?curr->left:curr->right;
if(curr!=root){
    if(curr=parent->left)
      parent->left=child;
    else
      parent->right=child;
  }else
    root=child;
    free(cur);}}
```

5. ADVANTAGES & DISADVANTAGES OF BST.

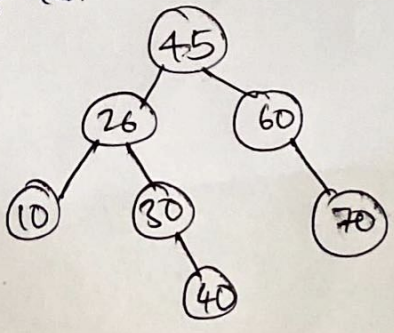
Advantages:

- BST is fast in insertion & deletion when balanced
- Very efficient & its code is easier than linked list.

Disadvantages:

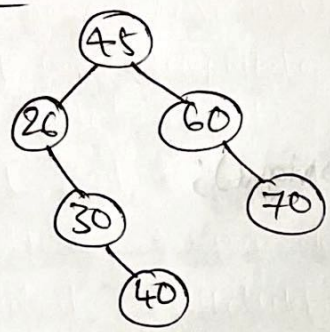
- Shape of the tree depends upon order of insertion.

6. Create a binary search tree - 45, 26, 10, 60, 70, 30, 40. Delete keys 10, 60, & 45.

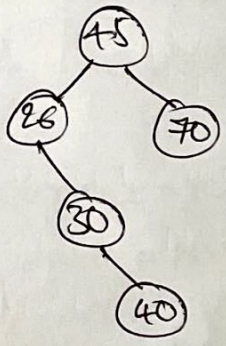


Deletion:

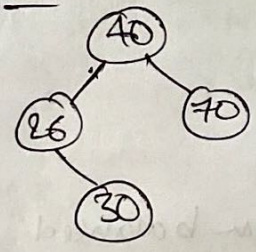
1. 10



2. 60



3. 45



RA2111030010051
T.ABHINAYA SREE

DSA ASSIGNMENT - 3

Consider a hash table of size 7 and a hash function $(3x+4) \bmod 7$. Assuming that hash table is initially empty, which of the following is the contents of the table when the sequence 1,3,8,10 is inserted into the table using closed hashing?

The correct option is 1, 10, 8 ---, 3

Size of hash table = 7

$h(x) = (3x+4) \bmod 7$

$h(1) = (3 \cdot 1 + 4) \bmod 7$

$= 7 \bmod 7 = 0$; insert 0 at 0th position

$h(3) = (3 \cdot 3 + 4) \bmod 7 = 13 \bmod 7 = 6$;

insert 6 at 6th location

$h(8) = (3 \cdot 8 + 4) \bmod 7 = 28 \bmod 7$

$= 0$; 0th position is already filled by element 3 so insert 8 at next free location which is 1st position.

$h(10) = (3 \cdot 10 + 4) \bmod 7 = 34 \bmod 7$

$= 6$

but 6th position is already filled with the element 3 so insert 10 at next free location which is 2nd position.

1, 10, 8, --- 3

0, 1, 2, 3, 4, 5, 6