[i]Name: Corn Daveat

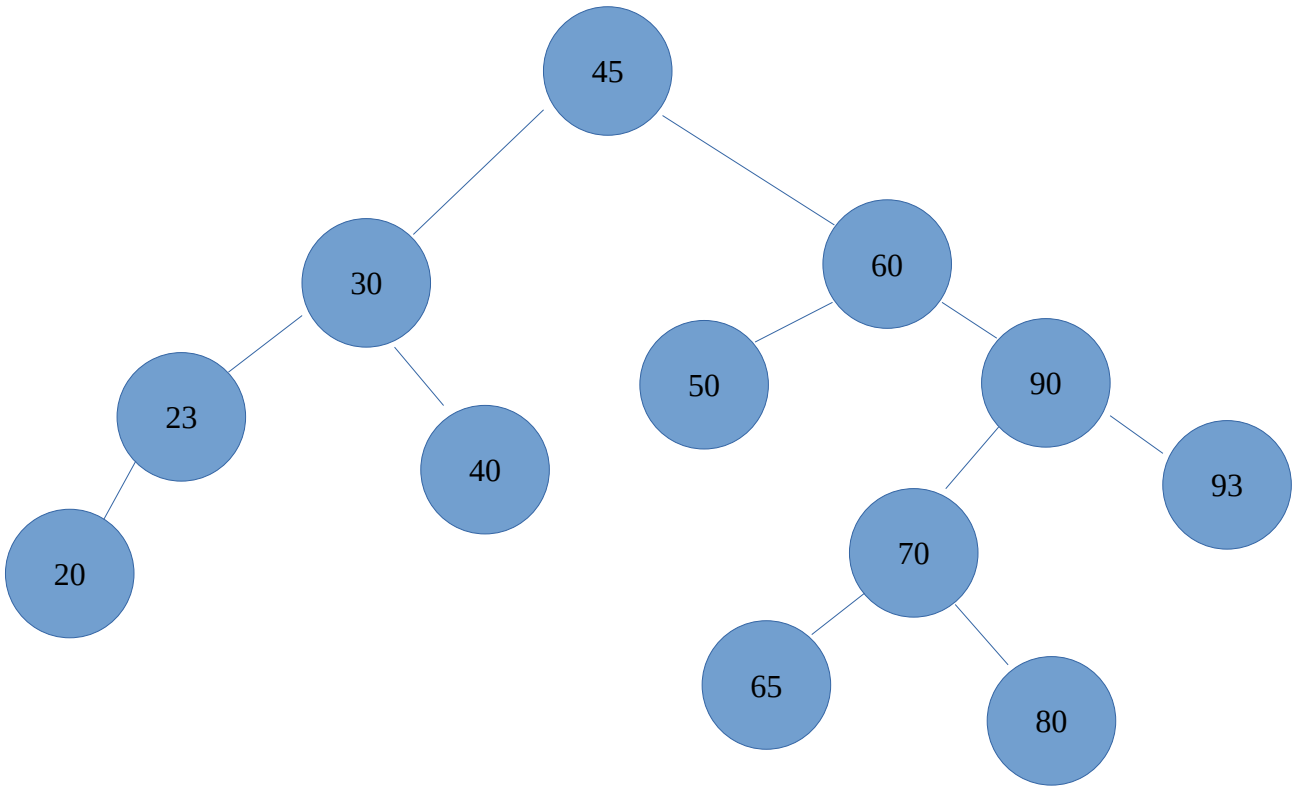
Class: M3

Exam

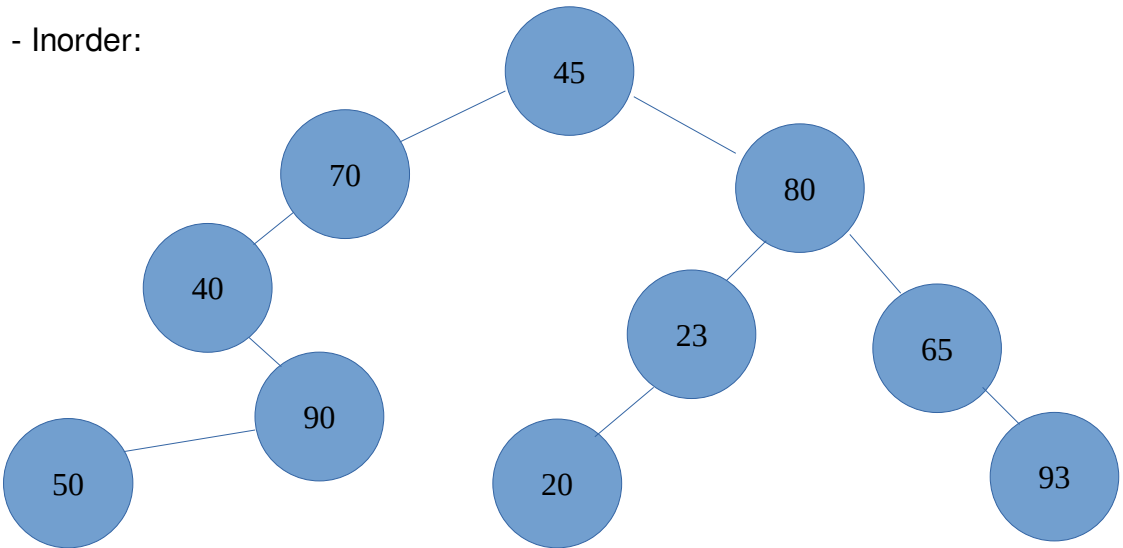
III.

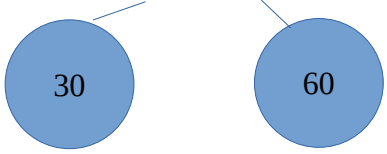
A. Binary Tree

- Preorder



- Inorder:





b.

Preorder:

45	30	60	23	40	50	90	20	70	93	65	80

Inorder:

45	70	80	40	23	65	50	90	20	93	30	60

c. សរសេរ Algorithm

```
struct Node* newNode(int data){
        struct Node* node = (struct node*)malloc(sizeof (struct Node));
        node→data = data;
        node→left = NULL;
        node→right = NULL;
        return node;
}
```

d.

II. Linked List:

1.

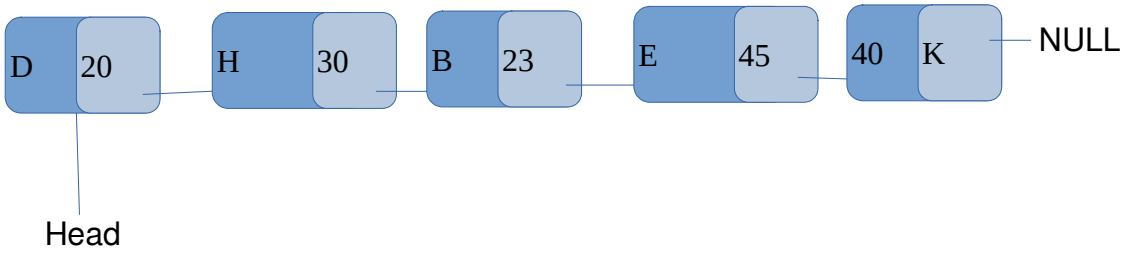
a.
```
void search(struct Node* employee){
        struct Node* ptr;
        for (ptr = front; ptr != NULL; ptr = ptr→next){
                if (ptr→salary < 120){
```

```
                ptr→salary = (ptr→salary * 15) / 100;
            }
        }
    }
```

b.

```
struct Node* insert (struct Node* nodeP, struct Node* nodePtr){
    if (head == NULL){
        head = nodeP;
        head→next = NULL;
        nodePtr = head;
    } else {
        nodeP→next = NULL;
        nodePtr→next = nodeP;
        nodePtr = nodePtr→next;
    }
    return tmp;
}
```

2. a.គូស Link List



b. C programming សម្តែងអោយ Available node:

```
struct Node* availableNode (){
        while(head != NULL){
                printf("Info %d\n", head→info);
                head = head→next;
        }
}
```

I.

- **a).** បង្កើតជា Adjacency Matrix សម្រាប់ Graph
  + A ->B->D
    B-> A->H
    D->B->K
    K->A->D
    H->F

  b). បង្កើតជា Adjacency List សម្រាប់ Graph
    A->B->D->K
    B->H->D->A
    D->A->B->K->H
    K->A->D->F
    F->H->K
    H->F->D->B