Lab9 Binary Search Tree (BST)

1. Main program

1) Menu 구성:

(1. Insert, 2. Delete, 3.Search, 4. Traverse 5. NumofNodes 6.Print 7.Quit)

2) For each command;

- **Insert:** "Enter number to insert: " gets input **Num**, **insert_tree**(Num)
- **Delete:** If (!tree_empty()) "Enter number to delete" gets input **Num**
  delete_tree(Num) else "**Tree is empty**"

- **Search:** if (!tree_empty()) "Enter number to search: " gets input **Num**
  temp=**search_tree**(root, Num)
  if (temp==NULL) "NOT found" else " number is found"
  else print "Tree is empty!!";

- **Traverse:** if (!tree_empty()) **levelorder( )** else "Tree is empty"
- **NumNodes:** If (!tree_empty()) **NumNodes**() else "Tree is empty"
- **Print:** **Draw_tree()**

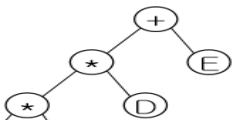
2. 알고리즘

-1), 2), 3) **Insert, Delete, Search**: (강의노트 참조)

Delete ➔ **FINDMAX 알고리즘 사용할 것.**

- **4) Traverse**: **// levelorder( )알고리즘 사용**

Levelorder: root level 부터 차례로 node print.



Ex)방문순서: + * E * D (root level 부터 level 씩 line by line 방문)

```
Procedure printlevel(Node *p, int level) {
  if (p==NULL) return false;
  if (level==1) print p->data; return true;
  else left=printlevel(p->left, level-1);
       right=printlevel(p->right, level-1);
       return left||right}
```

```
Procedure levelorder( ){
  int level =1;
  while(printlevel(root, level)) level++;
  }
```

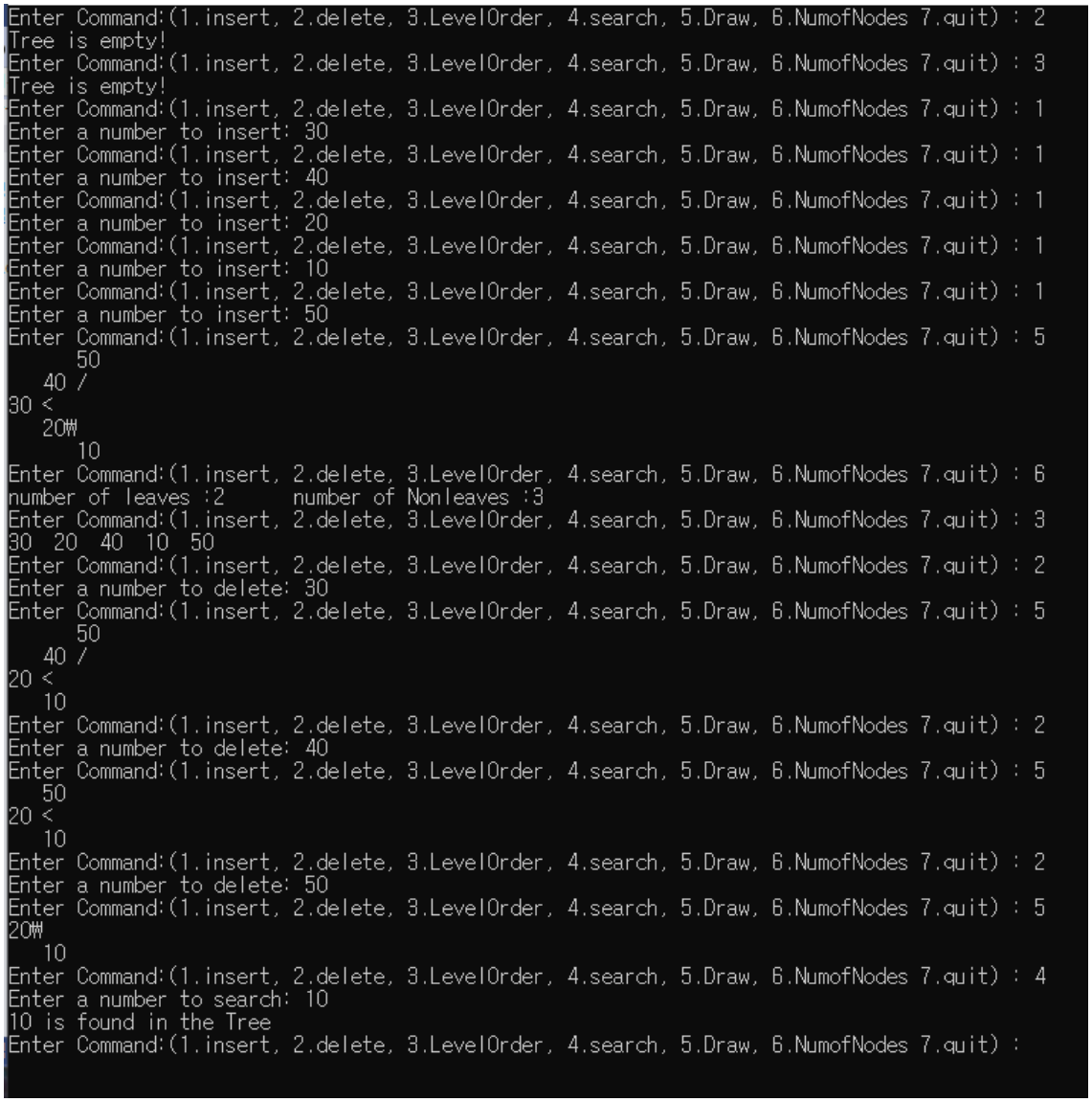
5) **Numofnodes:**

```
Procedure Numofnodes(){
   leaves(root, count);          print "number of leaves :
   Nonleaves(root, count);       print "number of Nonleaves :
}

Procedure   leaves(Node *p, int count) {   // if leaf nodes
  if (p) {     // if not NULL
        if (leaf)   count++;          //leaf => left && right child NULL
        else
           count = leaves(leftchild, count) + leaves(rightchild, count);
  else count = 0;
  return count;
 }

Procedure   Nonleaves(Node *p, int count) {   // if Non-leaf nodes
  if (p) {     // if not NULL
        if (leaf)   count =0;
        else   count = Nonleaves(leftchild,count) + Nonleaves(rightchild, count)+1;
    }
  else count = 0;
  return count;}
```

3. 테스트 절차(예):

1) Delete : Tree is empty
2) LevelOrder : Tree is empty
3) Insert : (30 40 20 10 50) 순서대로 insert 할것.
4) Print: tree 확인
5) LevelOrder:
6) NumNodes:
   Number of leaves: 1 number of Nonleaves: 2
7) Delete test:
   - Single 노드 테스트→ delete 20, DRAWTREE 로 출력
   - Leaf 노드 테스트 → delete 50, DRAWTREE 로 출력
   - 양쪽노드 테스트 → delete 30, DRAWTREE 로 출력
8) Print: tree 확인
9) Search: 30 "Not Found"

* Print: (강의 노트의 drawtree 함수 이용할 수 있음)


Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 2
Tree is empty!
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 3
Tree is empty!
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 1
Enter a number to insert: 30
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 1
Enter a number to insert: 40
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 1
Enter a number to insert: 20
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 1
Enter a number to insert: 10
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 1
Enter a number to insert: 50
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 5
50
40 /
30 <
20
10
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 6
number of leaves :2 number of Nonleaves :3
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 3
30 20 40 10 50
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 2
Enter a number to delete: 30
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 5
50
40 /
20 <
10
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 2
Enter a number to delete: 40
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 5
50
20 <
10
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 2
Enter a number to delete: 50
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 5
20
10
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) : 4
Enter a number to search: 10
10 is found in the Tree
Enter Command:(1.insert, 2.delete, 3.LevelOrder, 4.search, 5.Draw, 6.NumofNodes 7.quit) :