1번)

1-1)

1-2)

1-3)
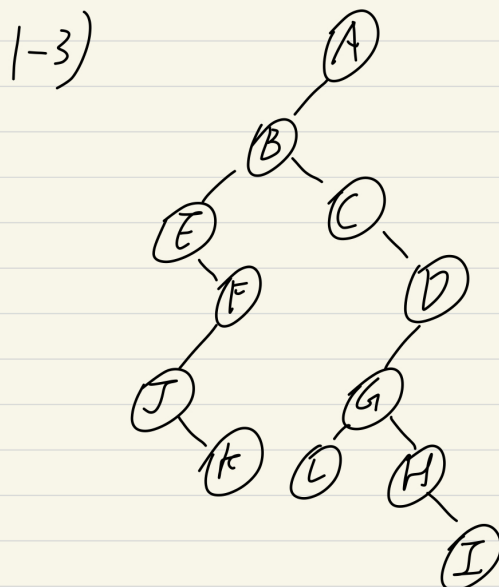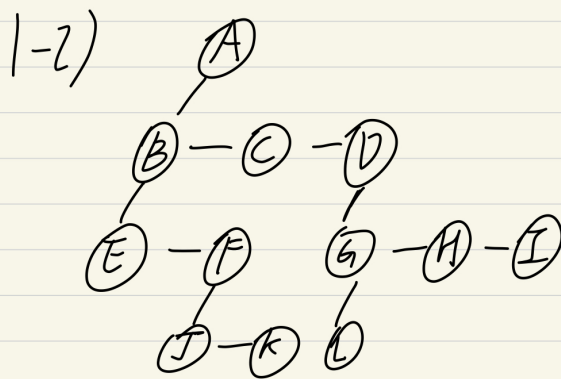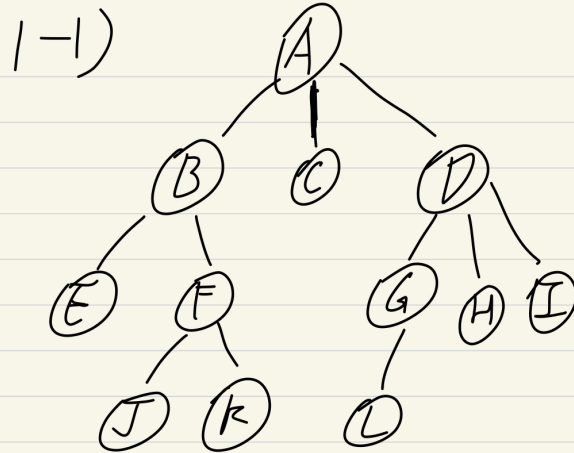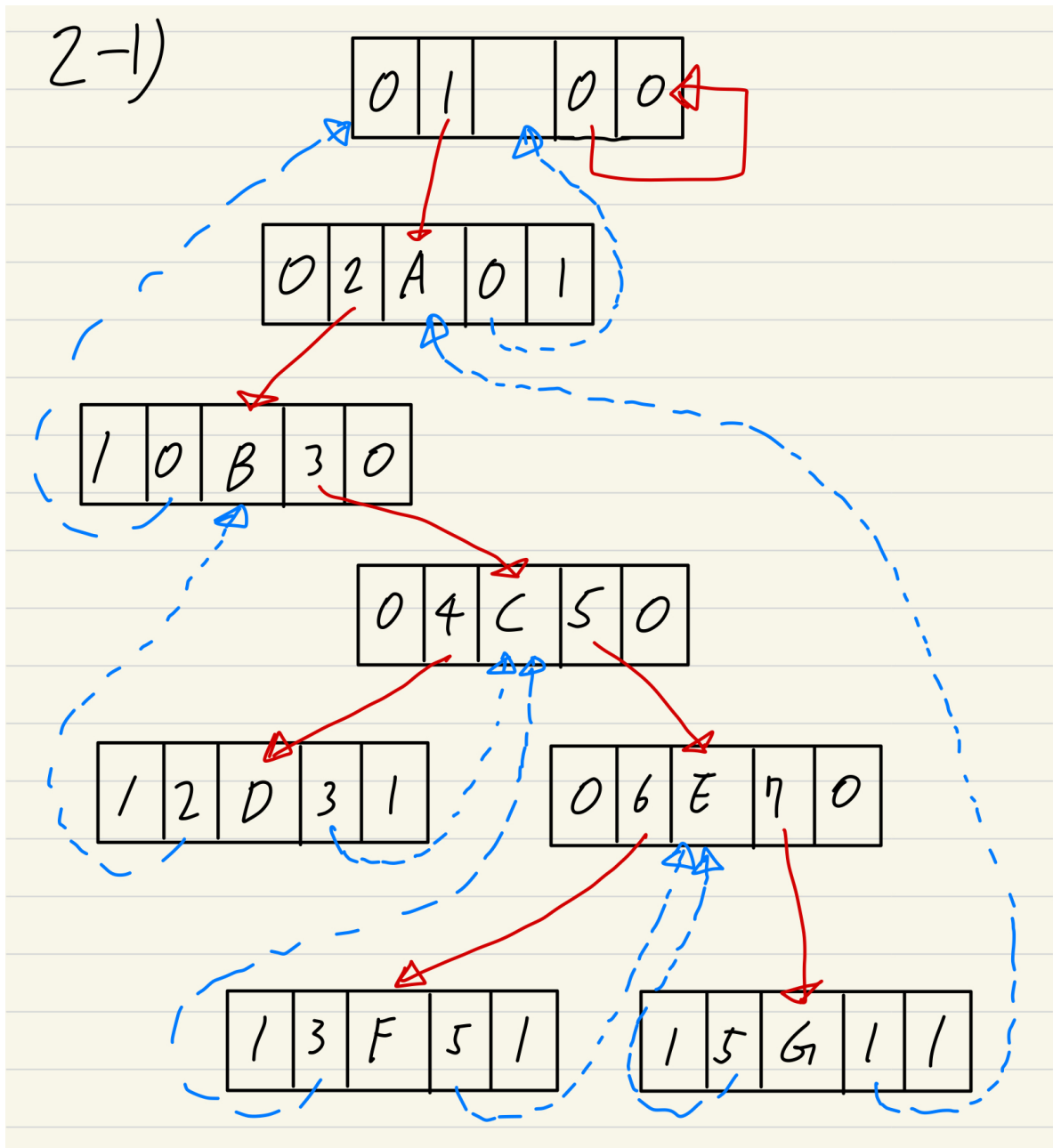
Inorder : E J K F B C L G H I D A

Preorder : A B E F J K C D G L H I

Postorder : K J F E L I H G D C B A

2번)

2-1)

| 0 | 1 | | 0 | 0 |

| 0 | 2 | A | 0 | 1 |

| 1 | 0 | B | 3 | 0 |

| 0 | 4 | C | 5 | 0 |

| 1 | 2 | D | 3 | 1 |

| 0 | 6 | E | 7 | 0 |

| 1 | 3 | F | 5 | 1 |

| 1 | 5 | G | 1 | 1 |

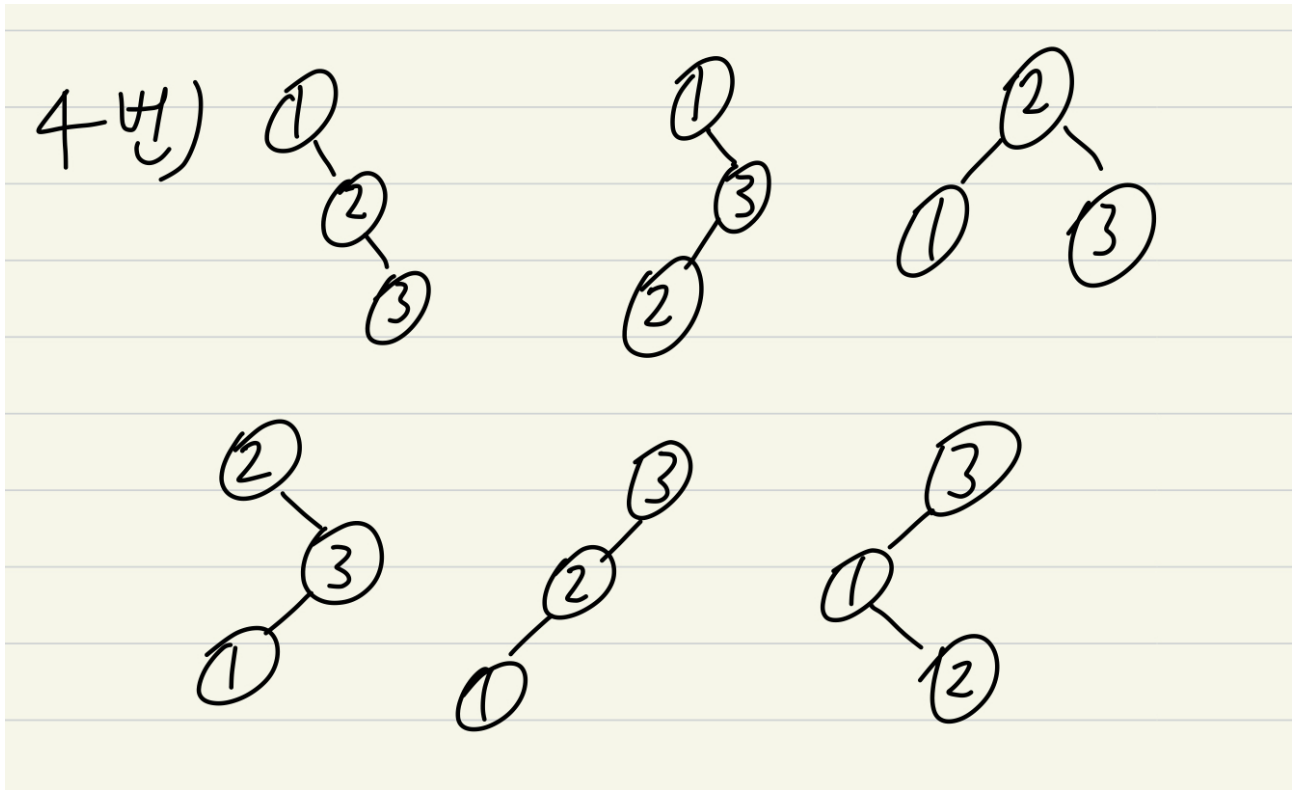2-2) B D C F E G A

2-3) A

3번)
 ABCEDGF

4번)

5번)

```
Node *Tree::insertBST(Node *ptr, int key)
{
    Node *p, *q;
    p = root;
    q = root;
    if (ptr == NULL) {
        ptr = new Node(key);
        ptr->data = key;
        ptr->left = NULL;
        ptr->right = NULL;
        if(!root)
            root = ptr;

    }

    while(p !=NULL){
        q = p;
        if(key < ptr -> data)
        p = ptr -> left;
        else p = ptr -> right;
    }

    if(key < q->data){
```

```
        Node *r = new Node(key);
        q -> left = r;

    }
    else{
        Node *r = new Node(key);
        q->right = r;
    }

}
```

6번)

```
int totalbstnode(Node* p) {
    int count = 0;
    if (p != 0) {
        count = 1 + totalbstnode(p->left) + totalbstnode(p->right);
    }
    return count;
}
```