CHIP 2학년 수업

중간고사 이전 복습 및 트리기초

2018년 1학기 5,6주차

목차

실습 1

실습 1-1

- https://www.acmicpc.net/problem/10828
- Baekjoon Online Judge – 10828번 문제
- Java나 Python같은 경우는 기본 라이브러리가 있지만, 가급적이면 C로 직접 스택을 구현해서 해보자!

실습 1-2

- https://www.acmicpc.net/problem/10845
- Baekjoon Online Judge – 10845번 문제
- Java나 Python같은 경우는 기본 라이브러리가 있지만, 가급적이면 C로 직접 큐를 구현해서 해보자!

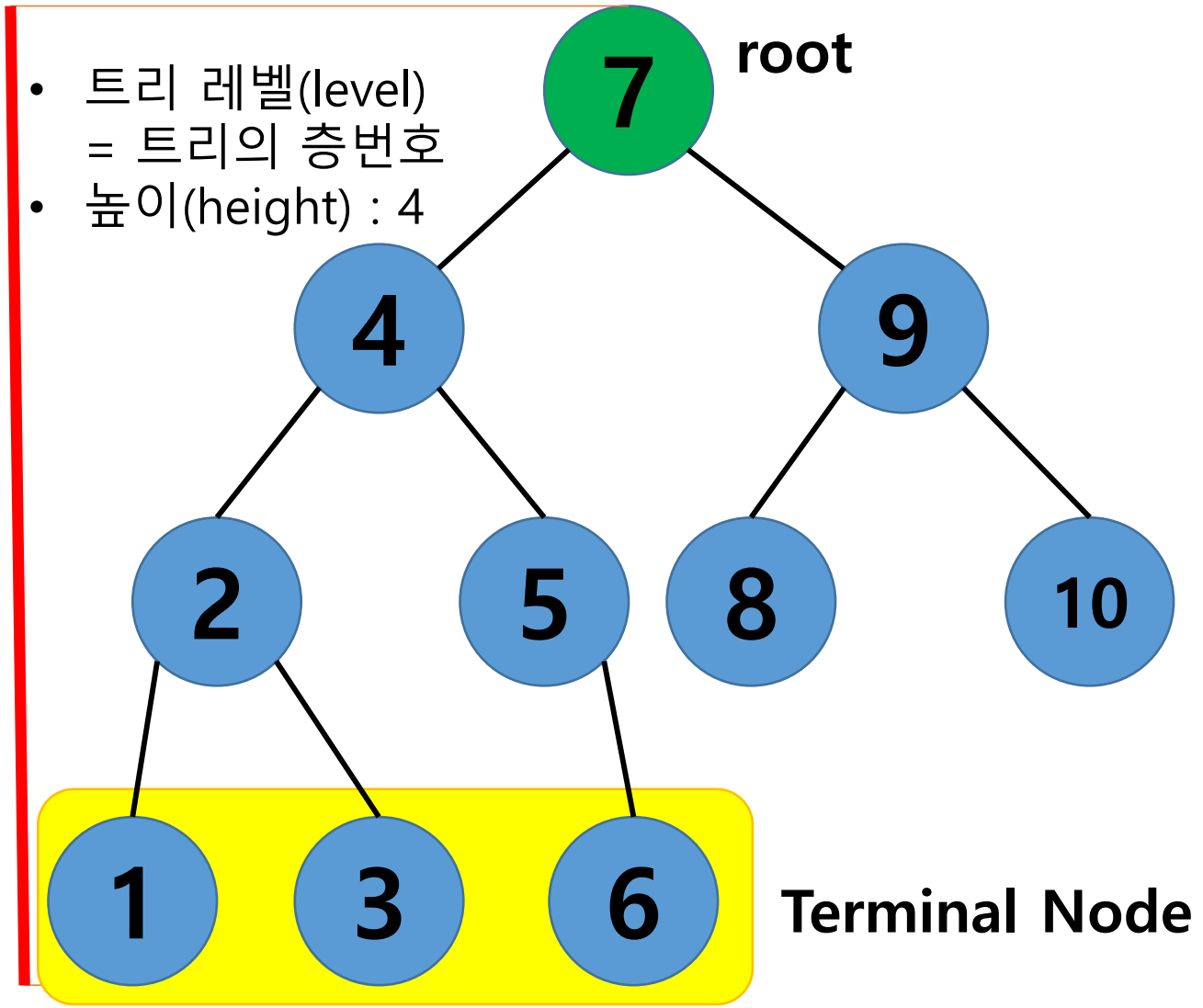
트리 개념

트리 개요

- 트리(Tree) 자료 구조
  - 계층적 구조를 가짐 (부모-자식 관계)
  - 루트 노드에서부터 시작
- 트리 응용
  - 파일 시스템 구조, Heap 구조, 탐색, 컴파일러 구문 분석 등
- 트리 종류
  - 대표적으론 이진트리가 있음(자식의 노드가 2개)
  - 그 외에 자식의 수가 3개 이상인 일반 트리도 있음

트리 용어

- 루트(root) : 7
- 단말노드(terminal Node) : 1,3,6
- 2의 형제노드 : 5
- 9의 자식노드 : 8, 10
  (차수(degree) :2)



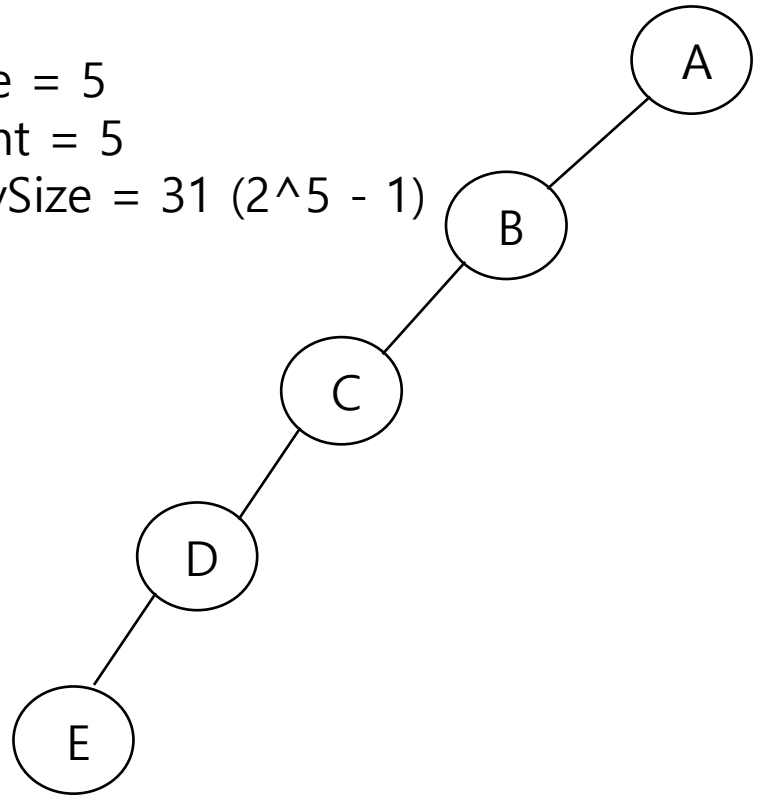
이진트리 개요

- 이진 트리의 정의
  - 많아봤자 최대 2개의 서브트리
  - 서브트리간의 순서 존재 = 여러 탐색이 가능

- 이진트리의 구조
  - 제한된 자식의 최대 수 = 2개
  - 다른 트리에 비해 비교적 간단함

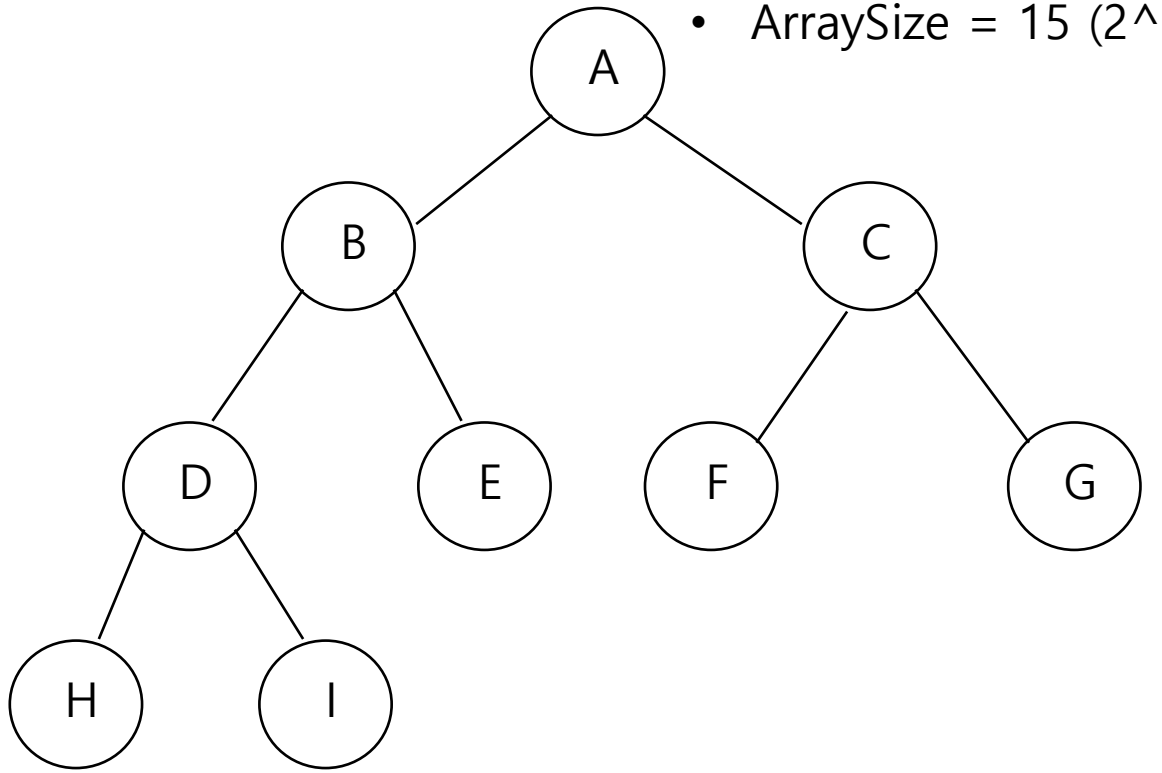
이진트리 종류

- Node = 5
- height = 5
- ArraySize = 31 (2^5 - 1)



skewed binary tree
(경사이진트리)

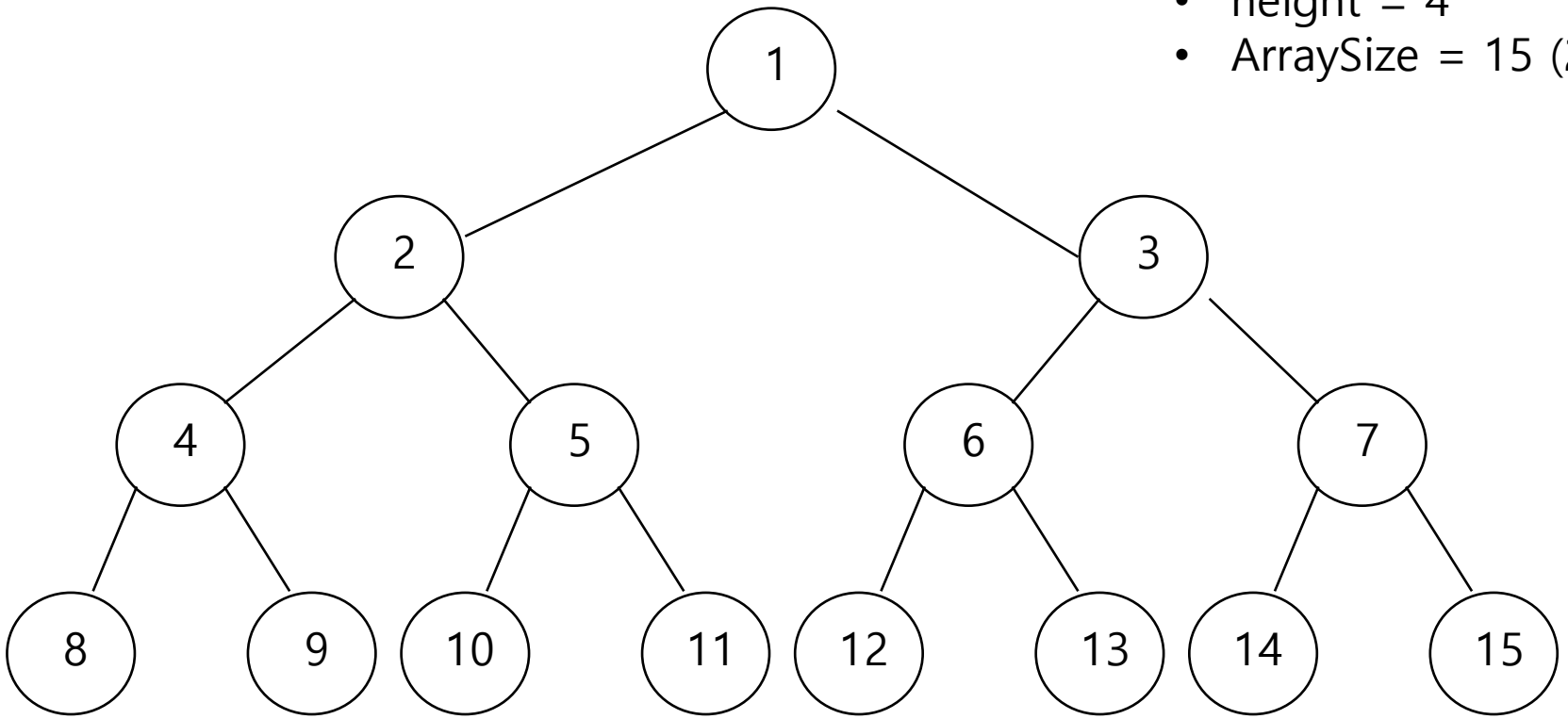
- Node = 9
- height = 4
- ArraySize = 15 (2^4 - 1)



complete binary tree(완전이진트리)

이진트리 종류

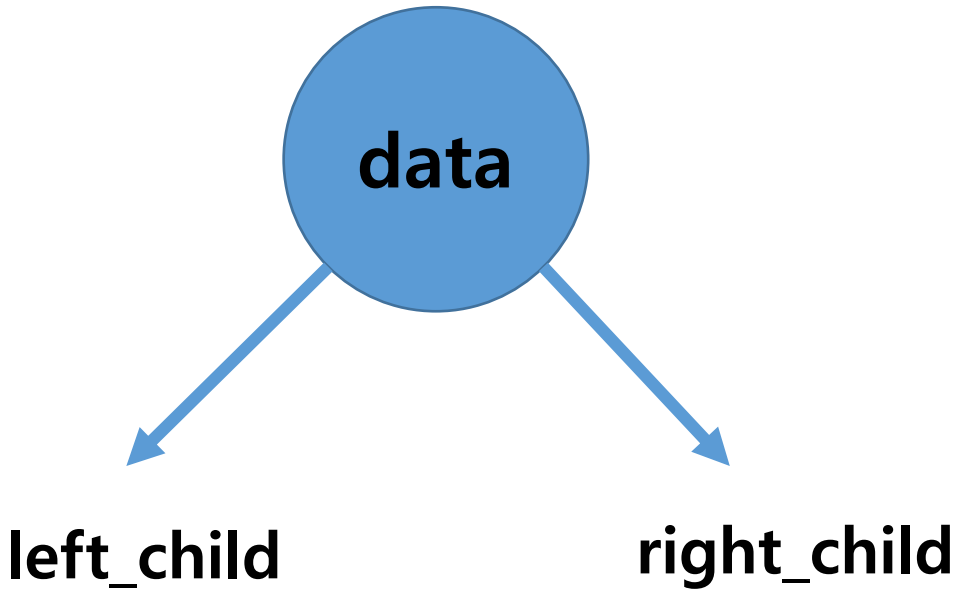
- Node = 15
- height = 4
- ArraySize = 15 (2^4 - 1)



full binary tree(포화이진트리)

이진트리 표현

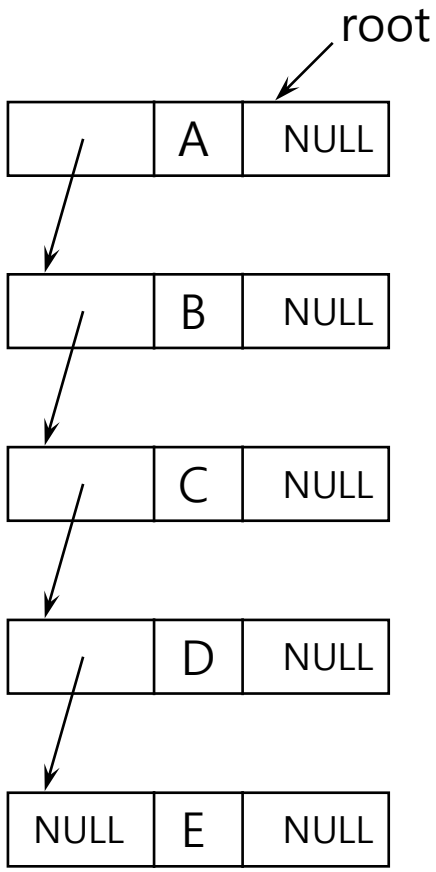
- 노드 구조



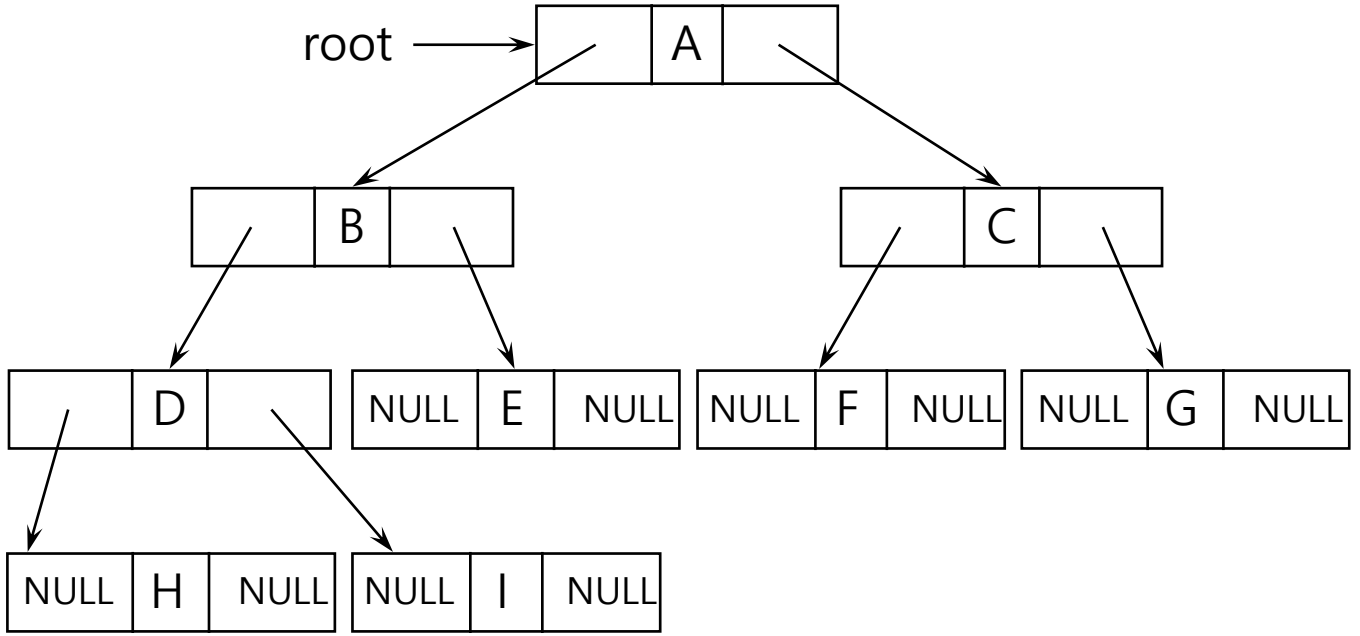
- 구조체 선언

```
typedef struct TreeNode{
    int data;
    struct TreeNode *left_child, *right_child;
} TreeNode;
```

이진트리 표현예시



skewed binary tree



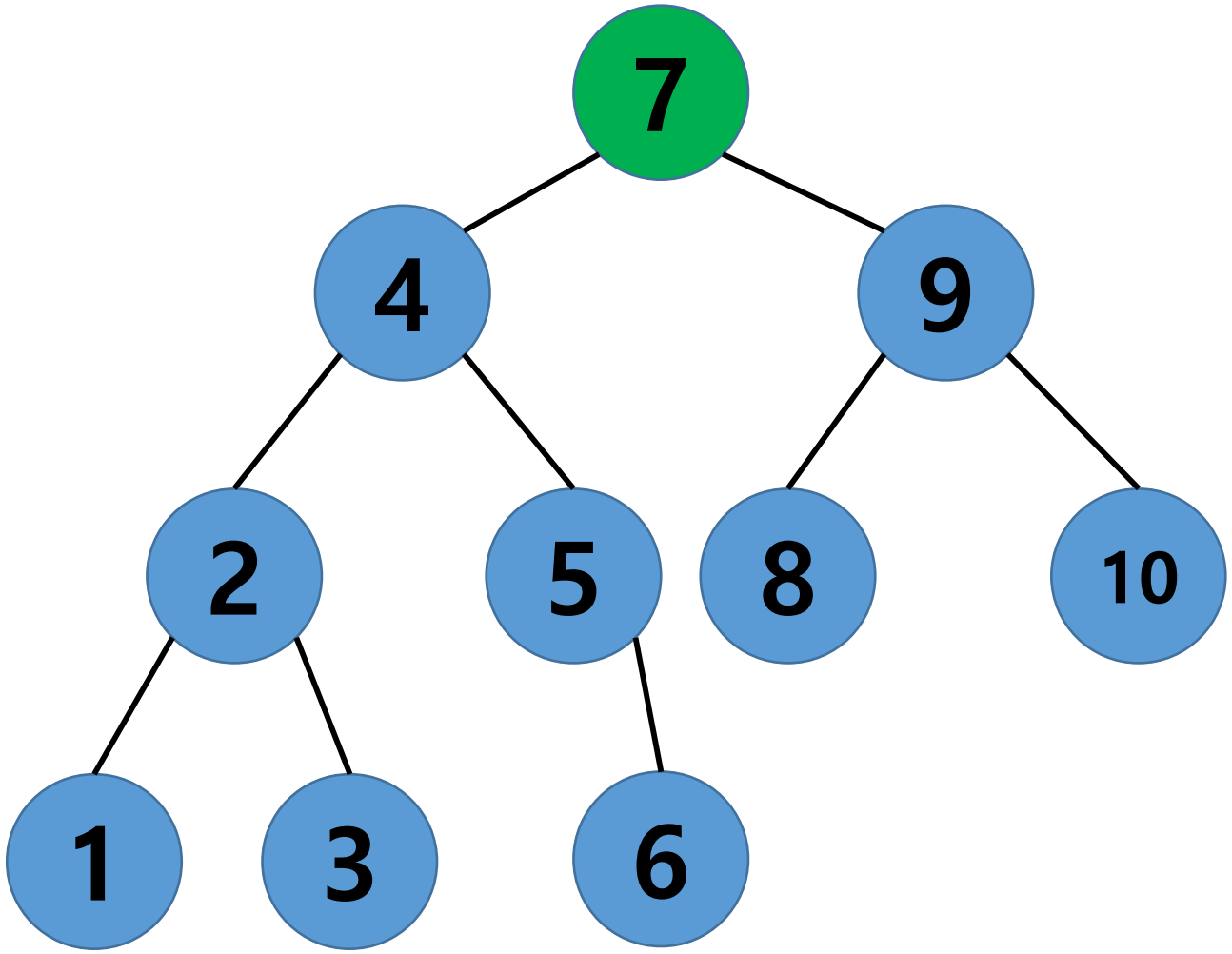
complete binary tree

이진트리 순회

이진트리 순회

- 순회방법
  - 전위순회(preorder traversal) : VLR
  - 중위순회(inorder traversal) : LVR
  - 후위순회(postorder traversal) : LRV
  - Level order traversal : Queue를 사용하는 순회

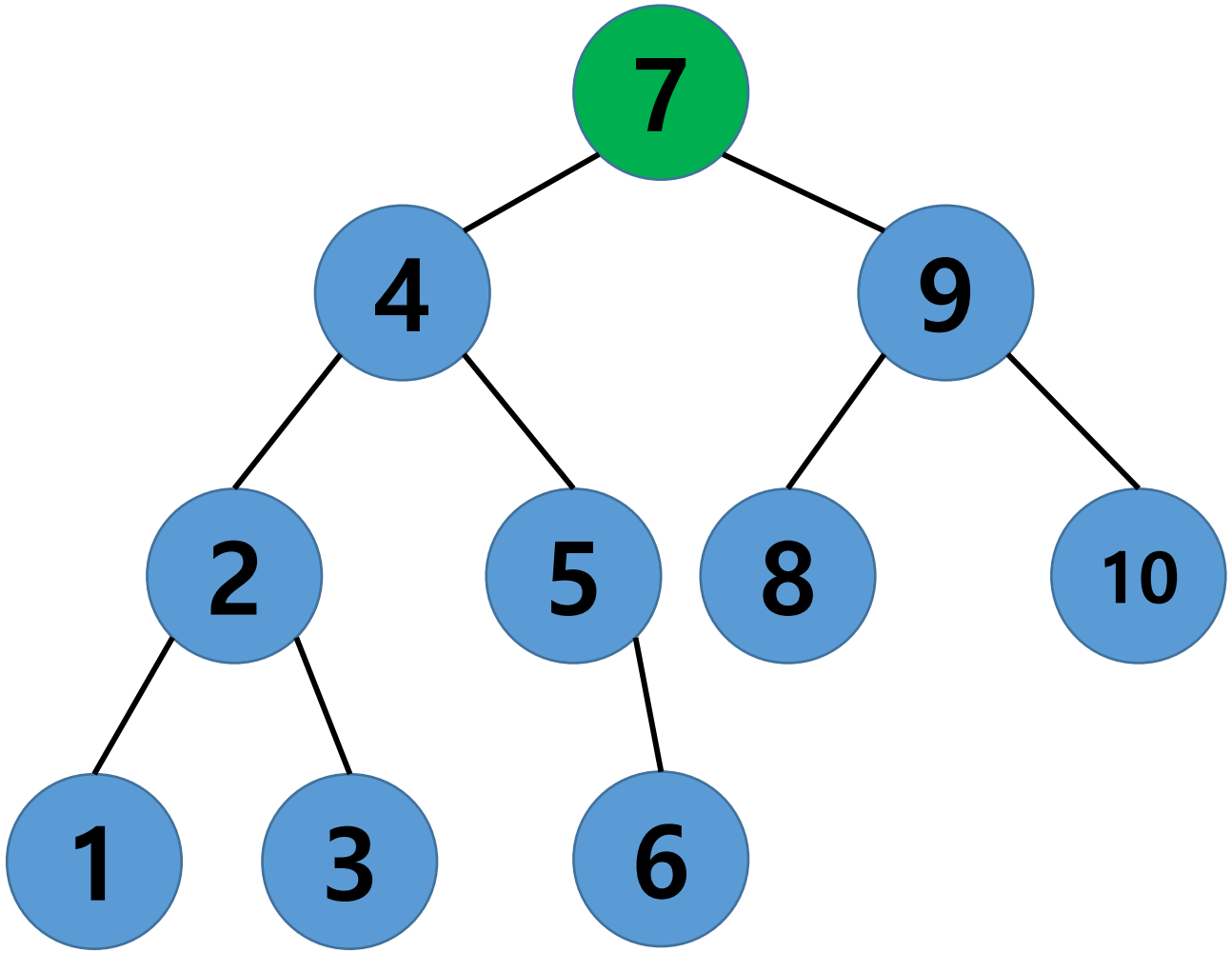
전위 순회(Preorder traversal)




```
preorder( TreeNode *root ){
    if ( root ){
      printf("%d-", root->data ); //V
      preorder( root->left);      //L
      preorder( root->right);     //R
    }
}
```

Output : 7-4-2-1-3-5-6-9-8-10

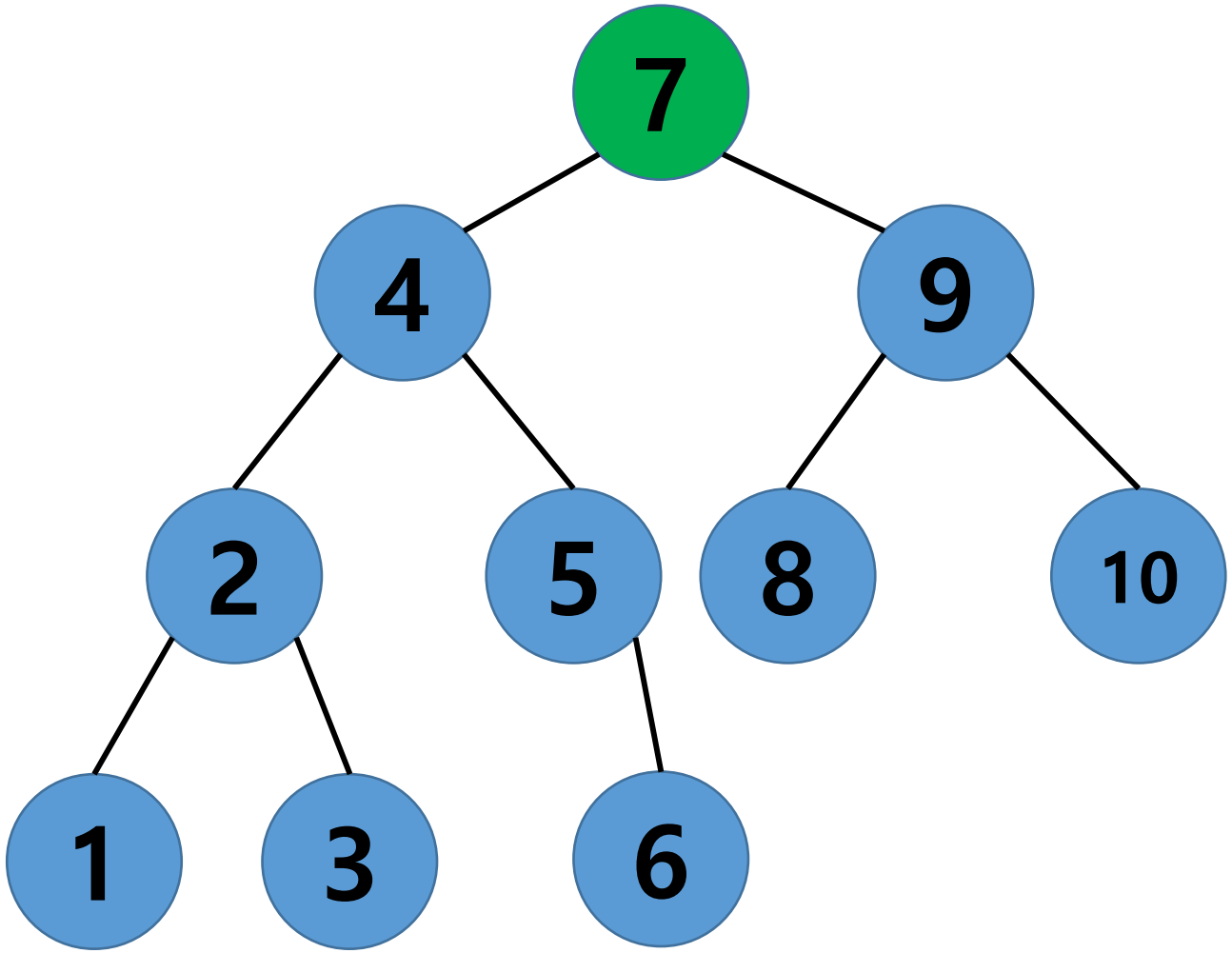
중위 순회(inorder traversal)




```
inorder( TreeNode *root ){
    if ( root ){
      inorder( root->left);          //L
      printf("%d-", root->data ); //V
      inorder( root->right);         //R
    }
}
```

Output : 1-2-3-4-5-6-7-8-9-10

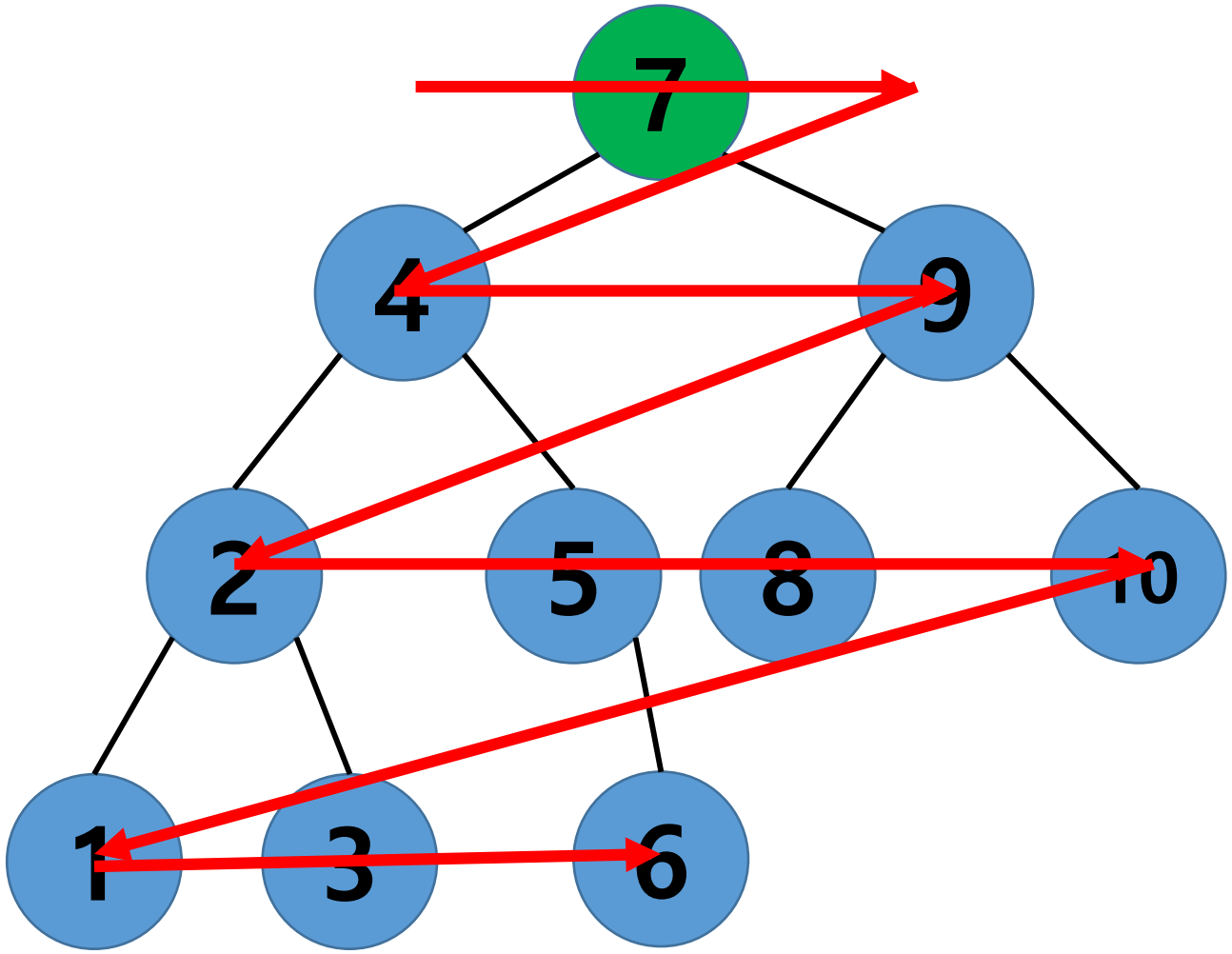
후위 순회(postorder traversal)




```
postorder( TreeNode *root ){
    if ( root ){
        postorder( root->left);      //L
        postorder( root->right);     //R
        printf("%d-", root->data); //V
    }
}
```

Output : 1-3-2-6-5-4-8-10-9-7

레벨 순회(Level order traversal)




```
void level_order(TreeNode *ptr)
 {
    QueueType q;
    init(&q);
    if (!ptr) return;
    enqueue(&q, ptr);
    while(is_empty(&q)) {
       ptr = dequeue(&q);
       printf("%d", ptr->data);
       if (ptr->left) enqueue(&q, ptr->left);
       if (ptr->right) enqueue(&q, ptr->right);
      }
   }
```

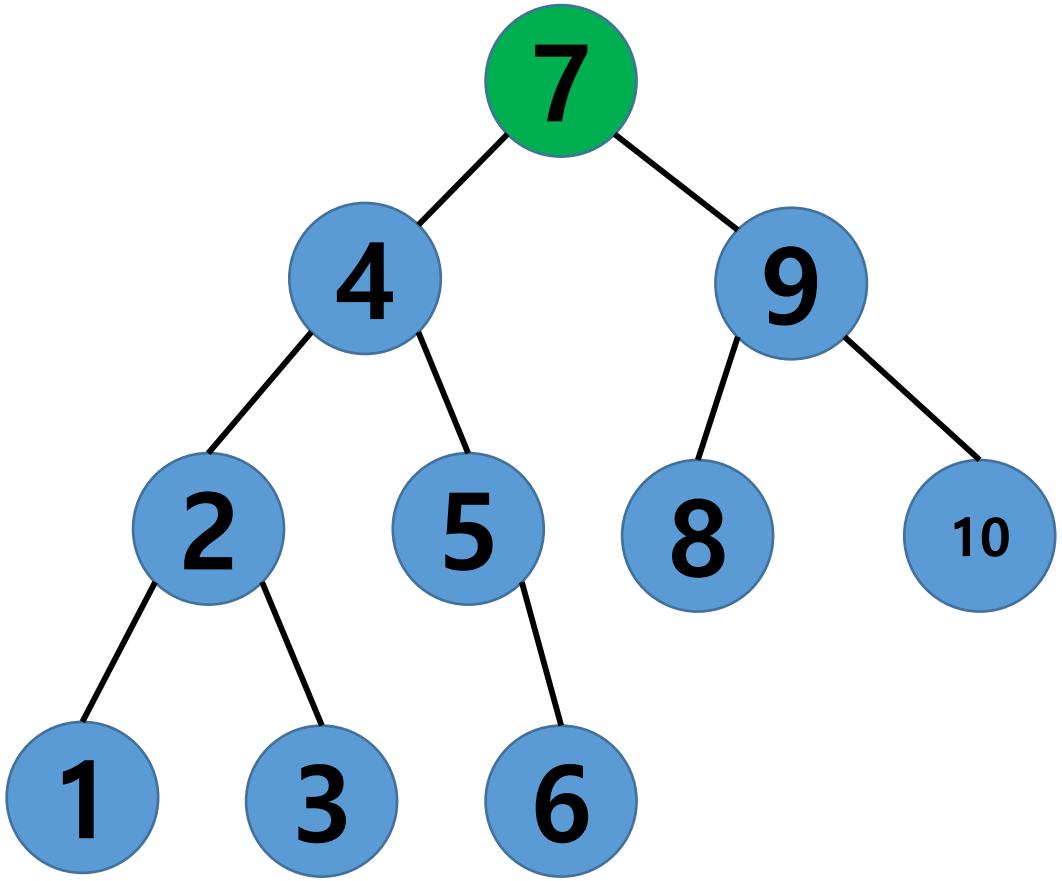
Output : 7-4-9-2-5-8-10-1-3-6

이진탐색트리 구현
(05.10~)

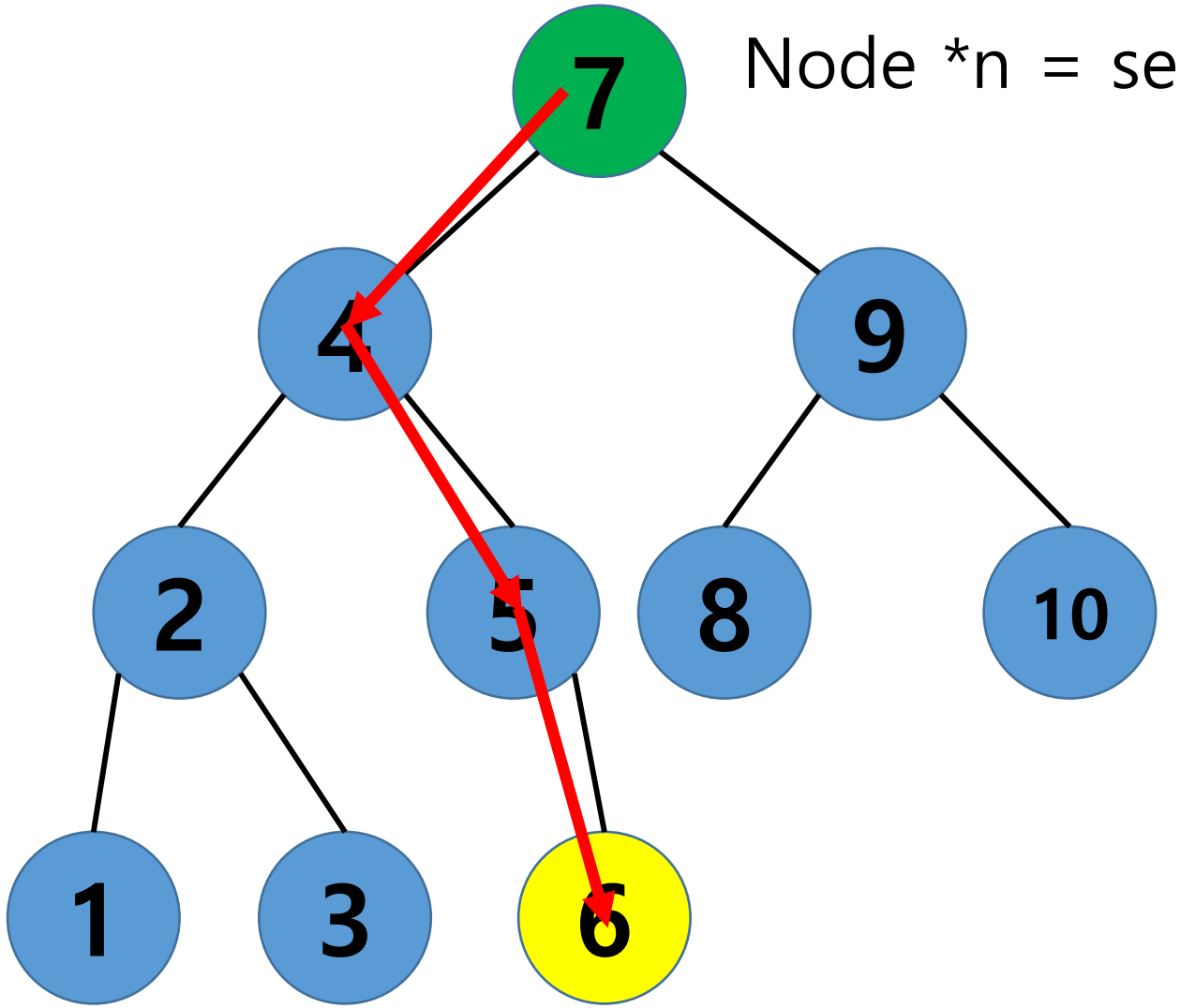
이진탐색트리 구현

- 이진 탐색 트리
  (Binary Search Tree : BST)

  - 모든 노드는 유일한 키 값 가짐
  - 왼쪽 서브트리 키값
    <= 한 노드의 키 값
    <= 오른쪽 서브트리 키값



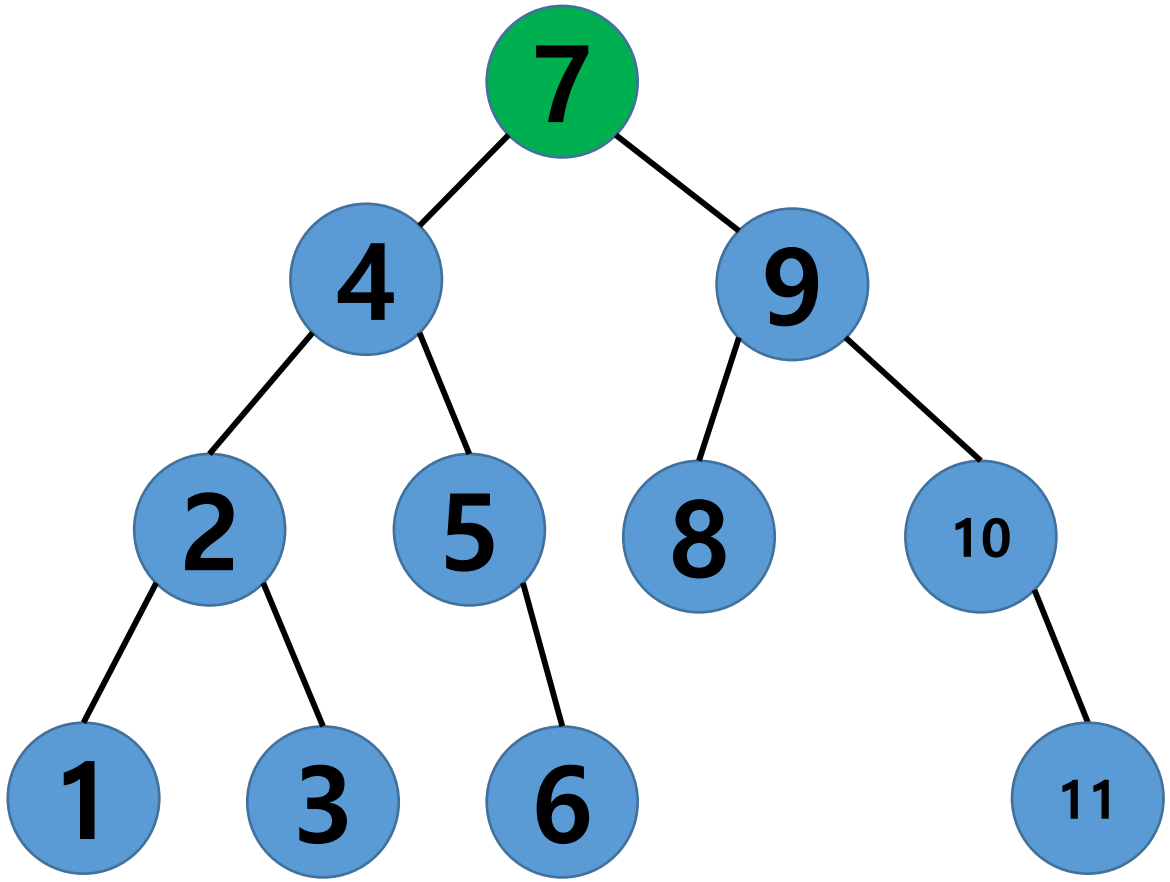
탐색(search) 연산



Node *n = search(root,6);

```
TreeNode* search_node(TreeNode *t, int key) {
    while( t != NULL && t->key != key) {
        if( t->key > key)  t  = t->left;
        else t  = t->right;
    }
    return t;
}
```

삽입(insert) 연산




```
new_node = (TreeNode *) malloc(sizeof(TreeNode));
new_node->key = 11;
new_node->left_child
    = new_node->right_child = NULL;
if (key < parent->key)
    parent->left_child = new_node;

else
    parent->right_child = new_node;
```

실습 2

실습 2

- https://www.acmicpc.net/problem/1991
- Baekjoon Online Judge – 1991번 문제
- 트리 함수 제대로 익힐겸 직접 써가면서 해보자.