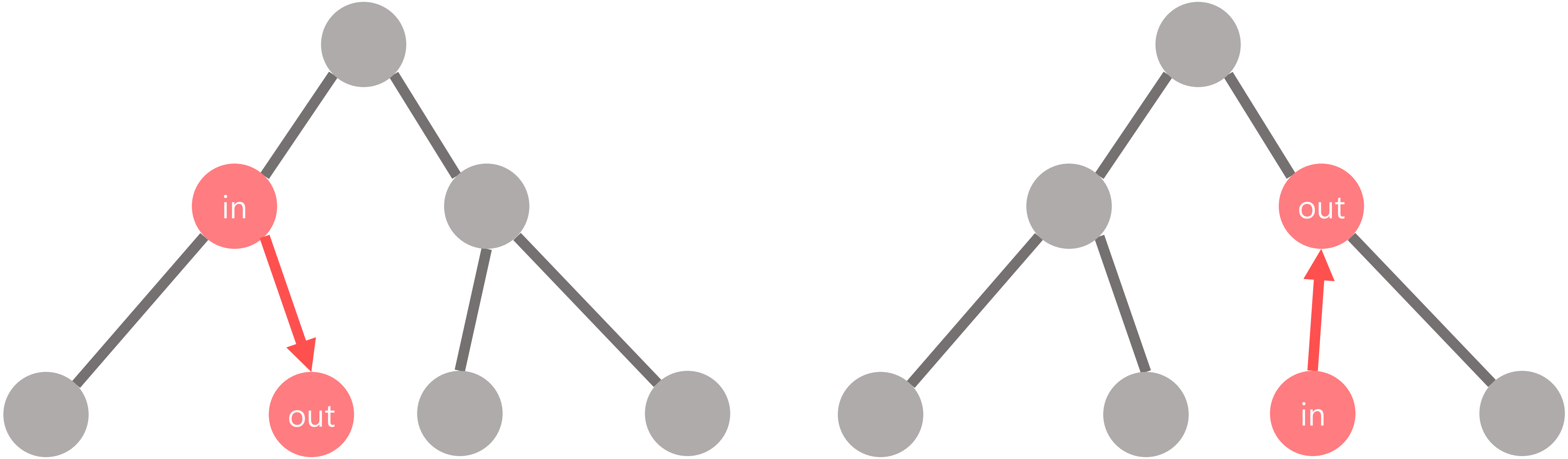
In-order successor

윤영교

문제

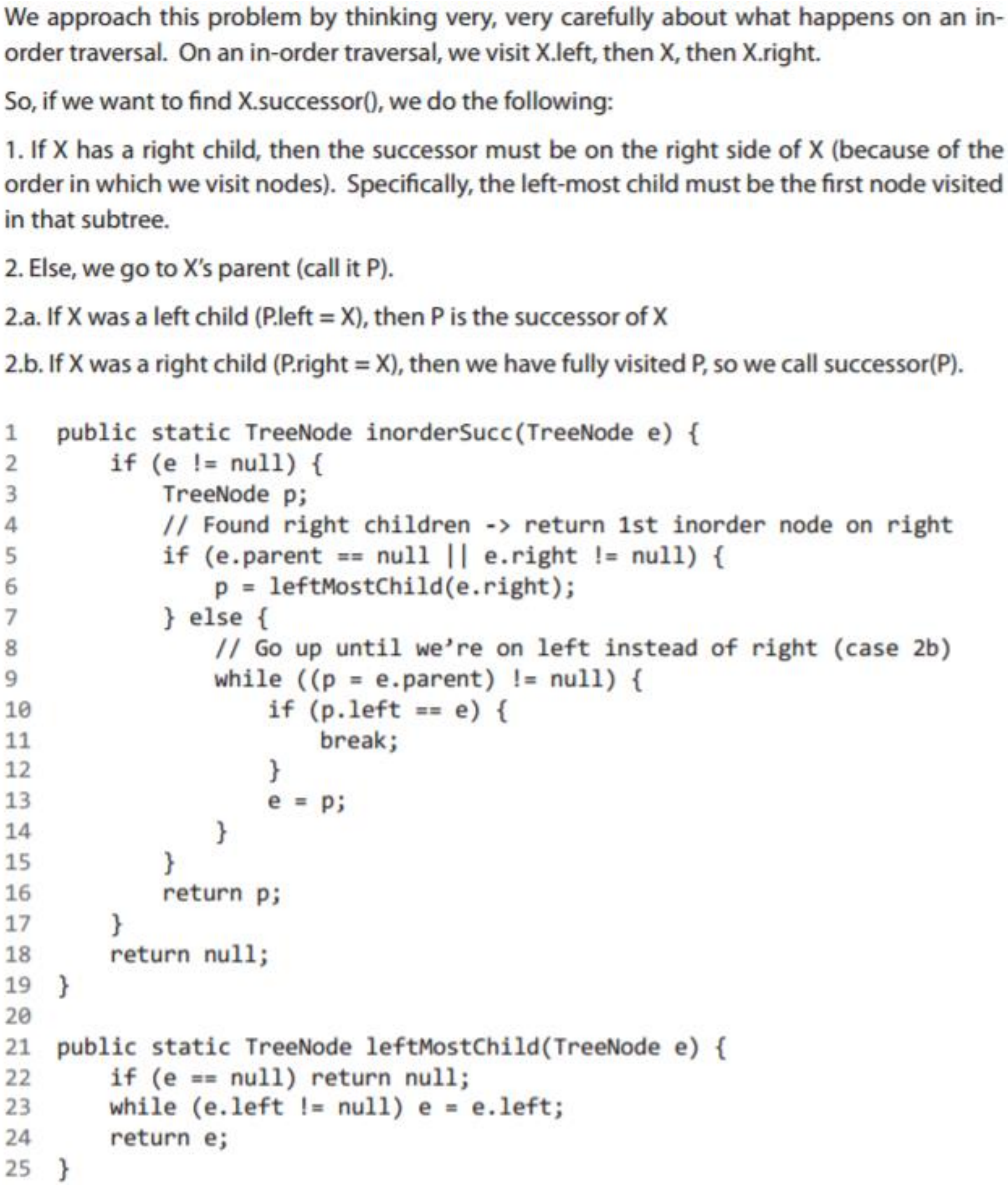
- 부모 노드로의 포인터를 가지고 있는 binary search tree가 있다고 하자. 이 트리에서 in-order traversal을 한다고 할 때, 주어진 노드에 대해서 다음으로 방문해야 할 노드를 구하는 알고리즘을 작성하라.
- (영문) Write an algorithm to find the ‘next’ node (e.g., in-order successor) of a given node in a binary search tree where each node has a link to its parent.

예시



답안

- 옆의 설명 참고



We approach this problem by thinking very, very carefully about what happens on an in-order traversal. On an in-order traversal, we visit X.left, then X, then X.right.

So, if we want to find X.successor(), we do the following:

1. If X has a right child, then the successor must be on the right side of X (because of the order in which we visit nodes). Specifically, the left-most child must be the first node visited in that subtree.

2. Else, we go to X's parent (call it P).

2.a. If X was a left child (P.left = X), then P is the successor of X

2.b. If X was a right child (P.right = X), then we have fully visited P, so we call successor(P).

```
public static TreeNode inorderSucc(TreeNode e) {
    if (e != null) {
        TreeNode p;
        // Found right children -> return 1st inorder node on right
        if (e.parent == null || e.right != null) {
            p = leftMostChild(e.right);
        } else {
            // Go up until we're on left instead of right (case 2b)
            while ((p = e.parent) != null) {
                if (p.left == e) {
                    break;
                }
                e = p;
            }
        }
        return p;
    }
    return null;
}

public static TreeNode leftMostChild(TreeNode e) {
    if (e == null) return null;
    while (e.left != null) e = e.left;
    return e;
}
```