

Data 31/3/2021

หาค่า Closest (กรณี n นานนนนน ~)

```
public class tree {
    Node root;

    public void printTree() {
        super.printTree(root);
    }

    public Node findClosest(int k) {
        Node current, closestNode;
        current = closestNode = root;
        int min_diff = Integer.MAX_VALUE;

        while (current != null) {
            if (k == current.key) return current;
            if (Math.abs(k - current.key) < min_diff) {
                min_diff = Math.abs(k - current.key);
                closestNode = current;
            }
            if (k < current.key) return current.left;
            else return current.right;
        }
    }
}
```

Tree Configurations for deleting node N

- ถ้า node ที่ลบมีลูก 2 คน ให้ Find Min.right (ในรูป x.key) *ถ้าเป็น leaf node ก็ลบทิ้ง!!*
- $N.key = x.key$
- ลบ x ใน right sub tree

Find the K^{th} (smallest) node

- ใช้ DFT หรือ return ต้นไม้ทั้งต้น $O(n)$ ถ้าไม่ไหวคือ... *ลองดู!!*
- $current = root$
- $int l = size(current.left)$
- if $k = l+1$ return root node
- if $k < l+1$ recursive ที่ left sub tree
- if $k > l+1$ เปลี่ยน $k = (k - l - 1)$ recursive ที่ right sub tree

FindNext

- $n = findMin(current.right)$
- ถ้าหาเจอ แล้วส่งให้ return null
- ถ้าไม่เจอ right sub tree?
- ไปหาคู่ไปเจอ Parent ที่อยู่ด้านขวา

Node FindNext (Node node)

```
if (node.right != null)
    return LeftDescendant(node.right) // case 1
else return RightAncestor(node) // case 2
```

Case 1

```
if (node.left == null)
    return node
else return LeftDescendant(node.left)
```

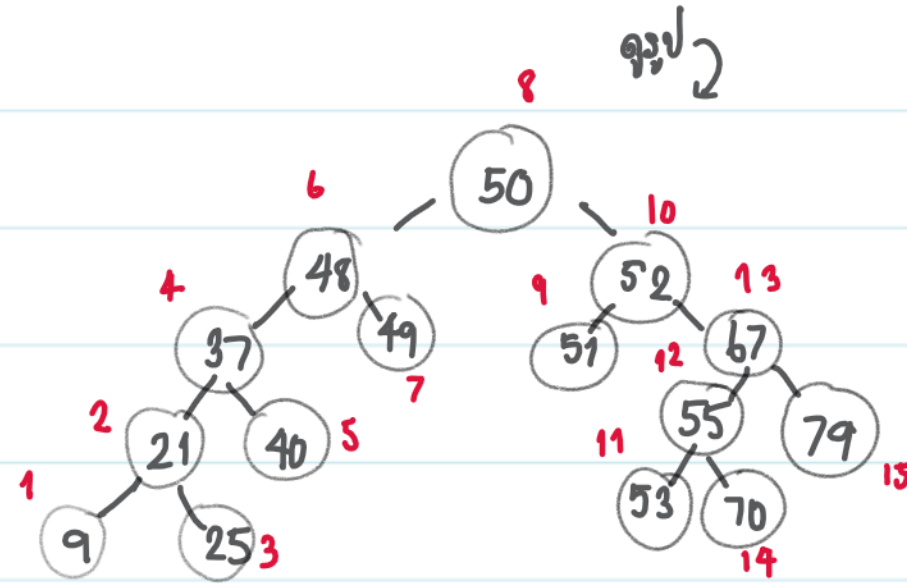
Case 2

```
if (node.key < node.parent.key)
    return node.parent
else return RightAncestor(node.parent)
```

* ถ้าใช้ while loop ว่าเป็นยังไงนะ?
ลองดูดู (อาจจะมีส่วนที่มันซับซ้อน)

อธิบายการหา K^{th} (smallest) Node แบบเข้าใจง่าย (มีนะ)

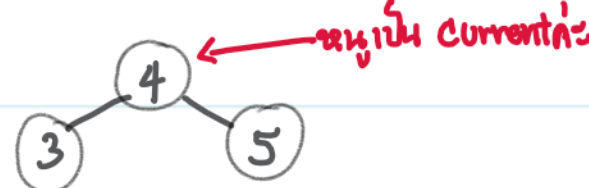
- แปลงตัวเลข Node ที่เล็กที่สุด
- แล้วใส่ใน array ตามลำดับ



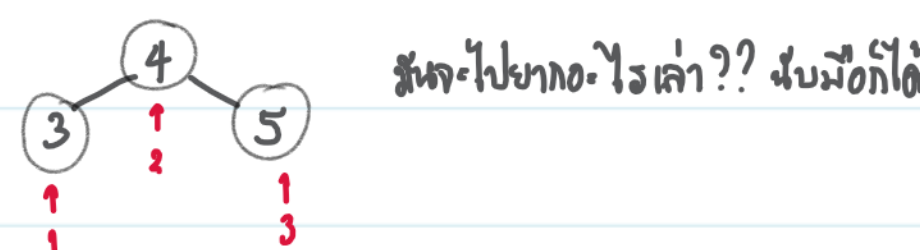
สัมพันธ์กับ DFT บน in order list
แต่ถ้าเป็น $O(n)$ ซึ่งไม่เร็ว... ไร้ว

หาด้วยวิธี ที่เป็น $O(\log N)$ ได้ไหม?

หาคู่ของ node $current = root$ node เดิม



หาคู่ของ node K^{th} ที่ 2



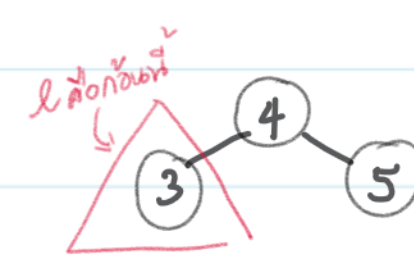
หาคู่ของ node ไม่ค่อยได้... ไร้ว?

1. หาคู่ int l คือ ขนาดของ left sub tree

2. หาคู่ k คือ ลำดับที่หาคู่

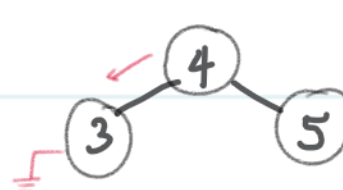
- ถ้า $k = l+1$ ให้ return root node

จากรูป $l = 1 \rightarrow 1+1 = 2$ เห็นไหม?



ถ้า k ที่หาคู่ไม่ได้... $l+1$

- ไปหาคู่ของ node เดิม

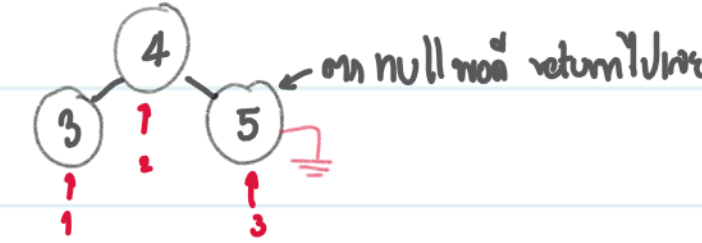


อธิบาย Find next แบบเข้าใจง่ายกว่า

- แปลงตัวเลข Node ที่เล็กที่สุด

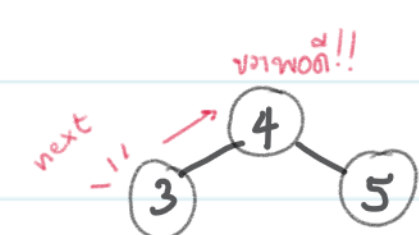
- หาคู่ n ตามลำดับที่หาคู่ใน right sub tree

- หาหาคู่ของ node null

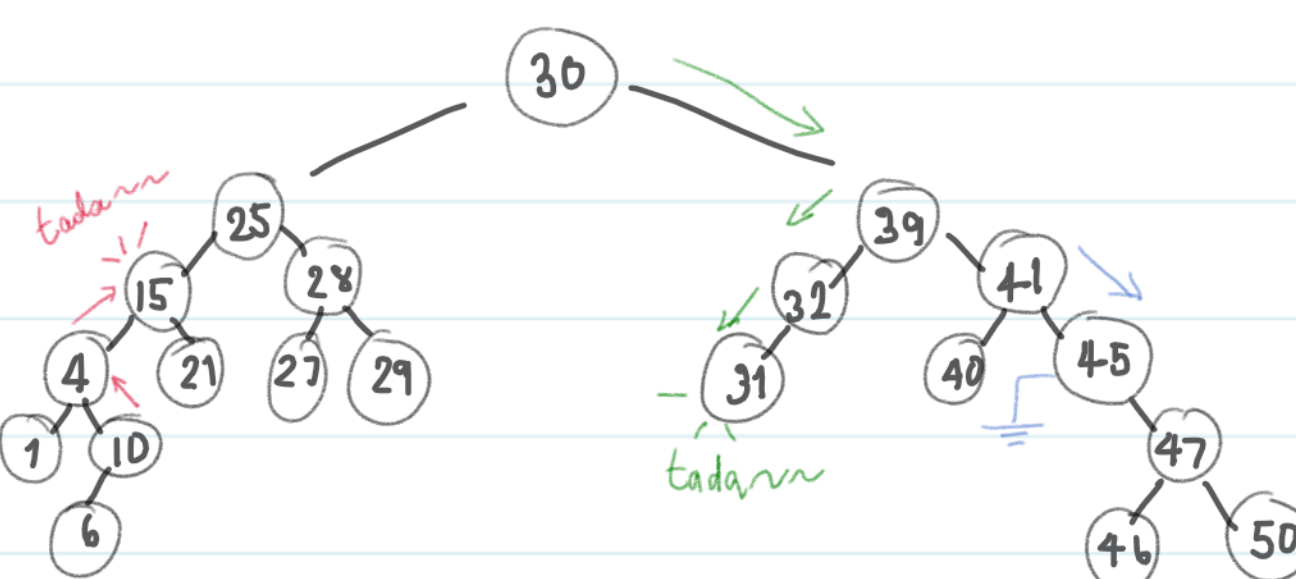


- ถ้าไม่เจอ right sub tree

ไปหาคู่ของ node เดิม



- หาหาคู่ของ node เดิม



• หา Next ของ node 10 : ไม่เจอ right sub tree \rightarrow ไปหาคู่ของ node เดิม

• หา Next ของ node 30 : มี right sub tree \rightarrow หาหาคู่ของ node เดิม

• หา Next ของ node 41 : มี right sub tree \rightarrow หาหาคู่ของ node เดิม