

# 중간고사 과제

Due: 5월 10일 18시

**1. BST with duplicate keys:** BST에서 duplicate key를 허용하려고 한다. 아래의 각 경우에 대해 Insert, Delete, Find 함수를 구현하시오.

## Parameter 제약 조건

- BST의 input parameter K는 int형으로 key값이 입력된다.
  - K의 범위는 1이상, 10000 이하이다.
  - K의 개수는 1이상, 1000 이하이다.

- A. 가장 간단하게는 각 node에 duplicate key의 count를 나타내는 field를 추가할 수 있다. Count field를 유지할 수 있도록 각 함수(Insert, Delete, Find)를 구현하시오.
- B. BST에 사용되는 key는 같아도 각각 서로 다른 노드로 구별되어야 하는 경우가 있을 수 있다. 이럴 경우,
  - a. left subtree, right subtree에는 각각 strictly smaller, strictly larger key value를 갖는 노드만 저장하고, key값이 같은 노드를 추가적으로 하나의 리스트로 유지할 수 있다. (리스트의 크기는 최대 100개까지 같은 key를 갖는 노드를 저장할 수 있다.) 이 때 각 함수(Insert, Delete, Find)를 구현하시오.
  - b. Duplicate key가 insert되면 무조건 left subtree 에 추가하는 것으로 약속한다고 할 때 각 함수(Insert, Delete, Find)를 구현하시오.
  - c. 위의 b와 같이 한다면 tree가 한쪽으로 skew되어 height 가 커질 수 있다. 이를 보완하는 방법이 있겠는가? 방법을 제안하고 구현하시오.

## 2. Range Query (이 문제에서는 duplicate key가 없다고 가정)

- A. BST에 대해 두 개의 key 값 k1, k2에 대해 k1보다 크거나 같고 k2 보다 작거나 같은 모든 key value를 count 하는 함수 RangeSearch(T, k1, k2) 구현하시오.
- B. RangeSearch(T, k1, k2)의 수행에 필요한 시간이  $O(h)$ 가 되도록 tree node 에 추가적인 정보를 기록할 수 있다고 한다. 어떤 정보를 기록하면 되는가? Insert(T, k), Delete(T, k)와 함께 RangeSearch 함수를 구현하시오.

## 3. Union-Find

Union-Find 에서 트리의 height 대신에 tree의 크기, 즉, 노드의 개수에 따라 tree를 merge 한다고 하자. 각 subtree의 root에 그 subtree에 속한 노드 개수를 저장하고, 두 개의 tree를 union 할 때는 노드 개수가 작은 트리를 더 큰 트리의 root의 child 로 연결한다. 다음의 명제를 (수행된 Union의 개수에 따라) 수학적귀납법을 이용하여 증명하시오.

T는 “Union by size”를 이용해 만들어진 Union-Find tree이고, T에 속한 노드의 개수가 n, T의 height는 h라고 할 때  $h \leq \log_2 n$  이다.