AppData\Local\Temp\ed0bf411-0eb2-472a-a1c1-a50ecf69cbed_OSAP_003_7_최종보고서(소스코드 포함).zip.bed\src\insert.cc

```
/*
MIT License
This file is part of the INHA_OSAP_003_7 project.
Copyright (c) 2024 tbmyong

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

작성자: 장태환
작성일(파일 생성일): 2024-12-05
작성일(파일 최종 수정일): 2024-12-17
*/

/*
Node* InsertNode(Node* current_node, int key)
: key가 작으면 왼쪽으로, 크면 오른쪽으로 이동하며 새 노드 생성
  후 정보 갱신 및 균형 조정

void Insert(int key)
: InsertNode 호출 후 리턴되는 노드를 root로 설정하고
  높이와 깊이의 합 출력
*/

#include "../base/avl.h"

Node* AVL::InsertNode(Node* current_node, int key) {
  // current_node가 nullptr이면 새 노드 생성
  if (current_node == nullptr) {
    current_node = new Node(key);
    return current_node;
  }

  // 현재 노드보다 작으면 왼쪽으로
  if (key < current_node->get_key()) {
    Node* left_child = InsertNode(current_node->get_left(), key);
    current_node->set_left(left_child);
```

```
52      left_child->set_parent(current_node);
53    }
54    // 현재 노드보다 크면 오른쪽으로
55    else if (key > current_node->get_key()) {
56      Node* right_child = InsertNode(current_node->get_right(), key);
57      current_node->set_right(right_child);
58      right_child->set_parent(current_node);
59    }
60
61    updater_.Update(current_node);
62    balancer_.Balance(current_node);
63
64    return current_node;
65  }
66
67  void AVL::Insert(int key) {
68    Node* new_root = InsertNode(root_, key);
69    set_root(new_root);
70    std::cout << CalculateDepthHeightSum(new_root, key) << "\n";
71  }
```