AppData\Local\Temp\2f4dbd71-5905-43d2-9d4f-7e2ecad6a5ca_OSAP_003_7_최종보고서(소스코드 포함).zip.5ca\src\utilities\update_node.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: 이현진, 임시영
작성일(파일 생성일): 2024-11-26
작성일(파일 최종 수정일): 2024-12-17
*/

/*
<Update 기능>
void UpdateHeight() : 현재 노드의 높이를 max(left의 높이, right의 높이) + 1로
                      update한다.
void UpdateSubtreeSize() : 현재 노드의 크기를
                    left subtree의 크기 + right subtree의 크기 + 1로
                    update한다.
void Update() : UpdateHeight(), UpdateSubtreeSize()
*/

#include "../../base/update_node.h"

void UpdateNode::UpdateHeight(Node* current_node) {
  if (current_node == nullptr) {
    return;
  }

  /*
  왼쪽 자식 높이 계산, 왼쪽 자식 노드가 존재하면 높이를 가져오고 존재하지
  않으면 0으로 설정
  */
  int left_height = (current_node->get_left())
                        ? current_node->get_left()->get_height()
```

```
                        : 0;

  /*
  오른쪽 자식 높이 계산, 오른쪽 자식 노드가 존재하면 높이를 가져오고 존재하지
  않으면 0으로 설정
  */
  int right_height = (current_node->get_right())
                         ? current_node->get_right()->get_height()
                         : 0;

  // 두 노드 중 더 큰 높이로 최종 높이 계산
  int final_height = ((left_height > right_height) ? left_height : right_height)
                     + 1;

  // 계산된 최종 높이를 현재의 노드 높이로 설정
  current_node->set_height(final_height);
}

void UpdateNode::UpdateSubtreeSize(Node* current_node) {
  if (current_node == nullptr) {
    return;
  }

  int left_size = (current_node->get_left())
                      ? current_node->get_left()->get_subtree_size()
                      : 0;

  int right_size = (current_node->get_right())
                       ? current_node->get_right()->get_subtree_size()
                       : 0;

  current_node->set_subtree_size(left_size + right_size + 1);
}

void UpdateNode::Update(Node* current_node) {
  UpdateHeight(current_node);
  UpdateSubtreeSize(current_node);
}
```