

G:\내 드라이브\src AVL\src\average.cc

```
1  /*
2  MIT License
3  This file is part of the INHA_OSAP_003_7 project.
4  Copyright (c) 2024 tbmyong
5
6  Permission is hereby granted, free of charge, to any person obtaining a copy
7  of this software and associated documentation files (the "Software"), to deal
8  in the Software without restriction, including without limitation the rights
9  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
10 copies of the Software, and to permit persons to whom the Software is
11 furnished to do so, subject to the following conditions:
12
13 The above copyright notice and this permission notice shall be included in all
14 copies or substantial portions of the Software.
15
16 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
17 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
18 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
19 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
20 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
21 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
22 SOFTWARE.
23
24 작성자: 임시영
25 작성일(파일 생성일): 2024-12-05
26 작성일(파일 최종 수정일): 2024-12-05
27 */
28
29 /*
30 <Average 기능 구현>
31 Average()
32 : 주어진 노드의 서브트리에서 최소값과 최대값의 산술 평균을 정수로
33 반환합니다.
34 FindMin()
35 : 서브트리의 최소값을 가진 노드를 찾습니다. 입력된 노드를
36 기준으로 왼쪽 자식 노드로 이동하면서 트리의 최소값을 찾습니다.
37 FindMax()
38 : 서브트리의 최대값을 가진 노드를 찾습니다. 입력된 노드를 기준으로 오른쪽 자식
39 노드로 이동하면서 트리의 최대값을 찾습니다.
40 */
41
42 #include "../base/avl.h"
43
44 int AVL::Average(int key) {
45     Node* current_node = FindNode(root_, key);
46     // key 가진 노드가 없으면 0 반환
47     if (!current_node) return 0;
48
49     // 서브트리의 최소값과 최대값 확인
50     int min_value = FindMin(current_node)->get_key();
51     int max_value = FindMax(current_node)->get_key();
```

```
52  
53     return (min_value + max_value) / 2;  
54 }  
55  
56 Node* AVL::FindMin(Node* current_node) {  
57     // 왼쪽 자식 존재하는 동안 왼쪽 이동  
58     while (current_node->get_left()) {  
59         current_node = current_node->get_left();  
60     }  
61     return current_node;  
62 }  
63  
64 Node* AVL::FindMax(Node* current_node) {  
65     // 오른쪽 자식 존재하는 동안 오른쪽 이동  
66     while (current_node->get_right()) {  
67         current_node = current_node->get_right();  
68     }  
69     return current_node;  
70 }  
71
```