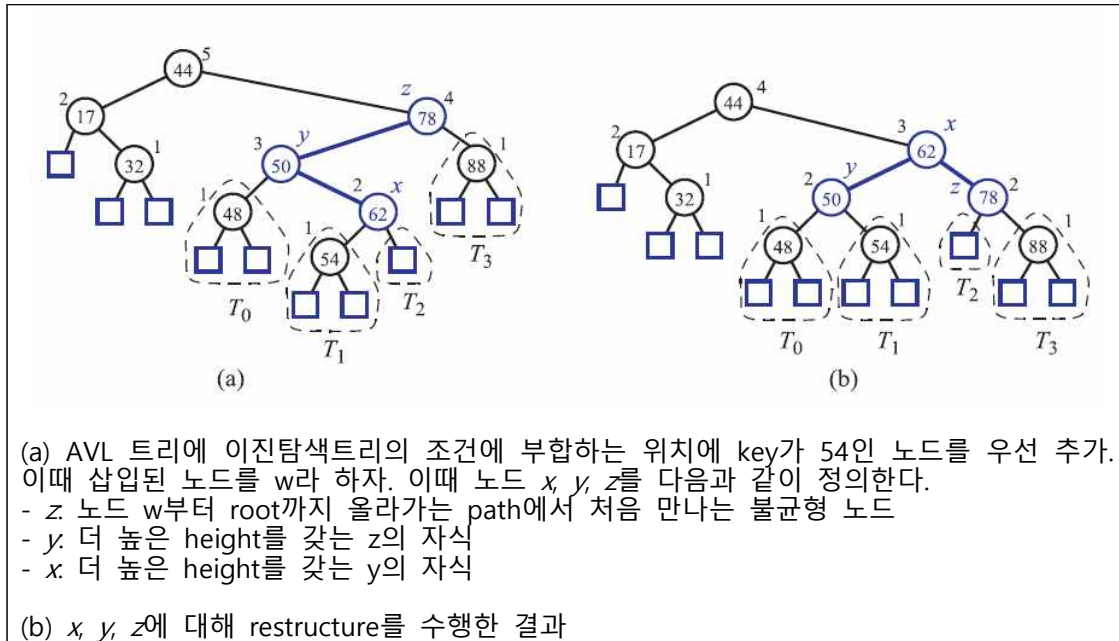


문제 번호	8주차-문제B
제목	균형이진탐색트리 구현
내용	<p>균형이진탐색트리인 AVL 트리 T는 다음의 높이균형속성(height-balance property)을 만족하는 이진탐색트리이다.</p> <ul style="list-style-type: none"> - 높이균형속성: T의 임의의 내부노드 v에서 v의 두 자식 간의 height가 많아야 1만큼만 차이난다. <p>이 속성을 만족하는 노드를 균형(<i>balanced</i>) 노드라 하고, 그렇지 않은 노드를 불균형(<i>unbalanced</i>) 노드라 한다.</p> <ul style="list-style-type: none"> • 단, 트리의 height는 root부터 leaf까지 가장 긴 path의 edge의 수로 정의한다. • 또한, 임의의 노드 u의 height는 u가 root인 서브트리의 height이다. <p>간단한 학생정보를 관리하는 AVL 트리를 구현한다. 학생정보 entry 형식은 다음과 같다.</p> <ul style="list-style-type: none"> - key: 유일한 학번 K ($10^5 \leq K < 2 \times 10^5$) - value: 이름 V (길이가 10을 초과하지 않는 문자열) <p>프로그램은 다음의 명령어를 수행한다.</p> <ul style="list-style-type: none"> • find K: key가 K인 노드를 찾으면 value와 height를 공백을 사이에 두고 출력한다. 그러한 entry가 없으면 -1을 출력 • insert K V: (key, value) 형식의 entry에 대해 삽입을 완료 후, 해당 entry 노드의 value와 height를 공백을 사이에 두고 출력 <p>위의 명령어들을 수행하는 프로그램을 제공된 main.cpp파일을 보완하여 완성하시오.</p> <p>사용할 수 있는 언어는 C, C++로 제한한다. 프로그램의 실행 시간은 3초를 초과할 수 없다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin 입력 속도를 개선할 수 있다.</p> <p style="text-align: center;">std::ios::sync_with_stdio(false)</p>

- insert 함수 그림 예제 (AVL 트리에 key 54인 노드를 삽입하는 예)



출처: Data Structures & Algorithms in c++

- restructure 함수의 의사코드

Algorithm restructure(x):

Input: A node x of a binary search tree T that has both a parent y and a grandparent z

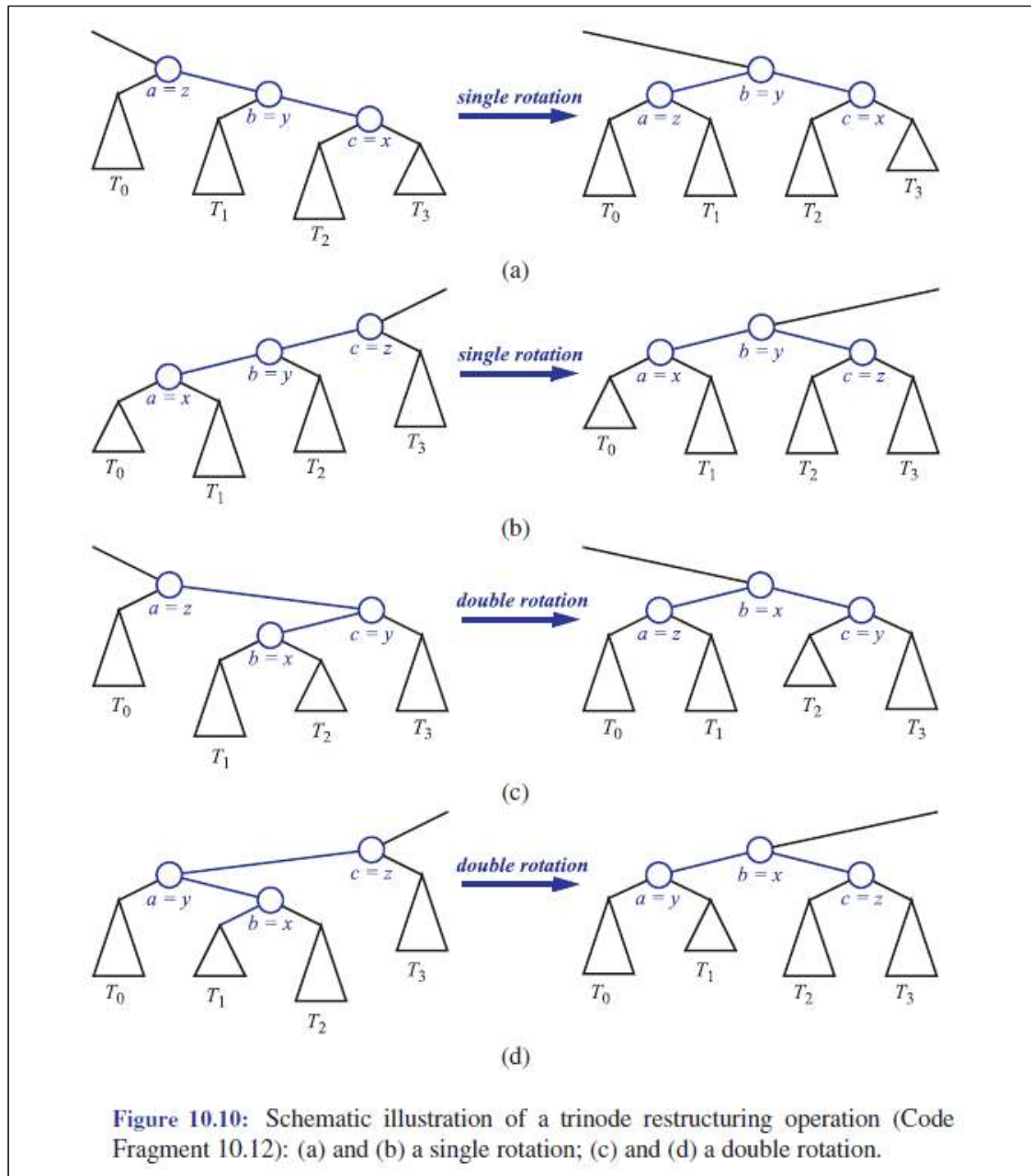
Output: Tree T after a trinode restructuring (which corresponds to a single or double rotation) involving nodes x, y , and z

1. Let (a, b, c) be a left-to-right (inorder) listing of the nodes x, y , and z , and let (T_0, T_1, T_2, T_3) be a left-to-right (inorder) listing of the four subtrees of x, y , and z not rooted at x, y , or z .
2. Replace the subtree rooted at z with a new subtree rooted at b .
3. Let a be the left child of b and let T_0 and T_1 be the left and right subtrees of a , respectively.
4. Let c be the right child of b and let T_2 and T_3 be the left and right subtrees of c , respectively.

Code Fragment 10.12: The trinode restructuring operation in a binary search tree.

출처: Data Structures & Algorithms in c++

- restructure 함수의 그림 예제



출처: Data Structures & Algorithms in c++

- 입출력 형식

입력 형식		<p>입력은 표준입력으로 다음과 같이 주어진다.</p> <ol style="list-style-type: none"> 1. 첫 번째 줄에 명령어의 수 N ($1 \leq N \leq 10^5$)이 주어진다. 2. 이후 N개의 줄을 통해, 처리할 명령어의 정보가 문제에서 설명한 형식으로 주어진다.
출력 형식		<p>출력은 표준출력으로 수행한다.</p> <p>각 명령어의 형식에 맞게 적절하게 한 줄 씩 출력한다.</p>
예	입력	<pre> 16 insert 165124 Euler insert 170674 Crome insert 158670 Euler insert 126285 Crome insert 119445 Crome insert 114192 Gauss insert 127643 Dijkstra insert 181315 Crome insert 175027 Amir insert 130363 Knuth insert 130633 Kepler find 130363 insert 120174 Aiden insert 153985 Aiden insert 111395 Ukkonen insert 113455 Tesla </pre>
	출력	<pre> Euler 0 Crome 0 Euler 0 Crome 0 Crome 0 Gauss 0 Dijkstra 0 Crome 0 Amir 1 Knuth 1 Kepler 0 Knuth 3 Aiden 0 Aiden 1 Ukkonen 0 Tesla 1 </pre>