

AVL Tree 구현

1. 구현 환경

- Window 10
- VS Community 2019, 16.8.4 버전

2. 제약 조건

- 입력은 정수만 받음
- 숫자 중복 insert 불가

3. 코드 명세

AVLTree.h

struct node	+int data;	key value
	+int height;	높이
	+node* left;	왼쪽 subtree ptr
	+node* right;	오른쪽 subtree ptr
	+node(int x);	constructor
class AVLTree	-node* root;	NULL로 초기화된 root
	-int n;	n=0: 아직 빈 tree
	-queue<int> q;	경로 출력을 위한 queue
	-bool find	tree 내에서 특정 key를 찾았는지 여부
	-int GetHeight(node* pNode);	pNode의 height return
	-node* rotateR(node* pNode)	pNode를 오른쪽으로 회전
	-node* rotateL(node* pNode);	pNode를 왼쪽으로 회전
	-void inorderPrint(node* pNode, ofstream* output);	tree 내의 모든 keys 오름차순 파일 출력
	-node* Insert(node* pNode, int x);	<ul style="list-style-type: none"> - (NULL일 시 node 생성 후) insert - balance 맞추기 - 이미 tree에 있는 key의 경우 안내 문구 표준 출력 ("Insertion failed: x already exists in the tree.")
	-node* Delete(node* pNode, int x);	<ul style="list-style-type: none"> - 해당 key를 찾아 delete - balance 맞추기 - 해당 key가 없을 경우 안내 문구 표준 출력 ("Deletion failed: There is no x in the key.")
	-node* Search(node* pNode, int x);	x를 찾으며 경로를 queue에 저장
	+void Insert(int x)	private Insert(root, x) 실행
	+void Delete(int x, ofstream* output)	<ul style="list-style-type: none"> - private Delete(root, x) 실행 - 삭제 한 x 파일 출력
	+node* Search(int x, ofstream* output)	<ul style="list-style-type: none"> - private Search(root, x) 실행 - 성공 시: queue에 저장된 x 경로 파일 출력 - 실패 시: 안내 문구 표준 출력("Searching failed: There is no x in the tree.")
	+void InorderPrint(ofstream* output)	tree 내의 모든 keys 오름차순 파일 출력

※성공한 insert/delete/search에 대한 출력은 모두 파일출력, 실패한 것에 대한 출력은 모두 표준출력으로 하였습니다.

main.cpp

- 1) "insert.txt" 입력 받아 tree에 insert
- 2) "insert.txt"에 있는 100만개의 keys 모두 search하여 "insert_result.txt"에 경로 저장
- 3) "delete.txt"에 있는 50만개의 keys 모두 삭제 후 "deleted_keys.txt"에 keys 저장
- 4) "remained_keys.txt"에 tree에 남은 keys 저장 후 해당 50만 개의 keys 모두 search 하여 "deleted_result.txt"에 경로 저장

4. 필요한 파일

- 1) "insert.txt" - 정수 100만개 목록
- 2) "delete.txt" - 삭제할 정수 50만개 목록

5. 실행 결과

<Insert> - <search> - <Delete> - <Search>의 순으로 자동 진행되며, 진행 사항은 표준 출력으로 안내됩니다. 프로그램이 종료된 시점에서는 다음 네 파일이 생성되어 있습니다.

- 1) "insert_result.txt" - tree에 insert된 keys의 search 결과 목록
- 2) "deleted_result.txt" - tree에서 delete된 keys 목록
- 3) "remained_keys.txt" - delete 이후 tree에 남은 keys 목록
- 4) "deleted_result.txt" - delete 이후 tree에 남은 keys의 search 결과 목록