RBTS(Red Black Tree for Searching)

[문제] 탐색을 위한 레드블랙트리(Red Black Tree for Searching)에 string 데이터를 추가(insert), 삭제 그리고 몇 가지 작업을 한다. 이 작업 도중에 관리하고 있는 RBTS의 depth k에 있는 node 또는 leaf node를 모두 찾아 왼쪽부터 오른쪽 순으로, 사전식 순서대로 모두 출력해서 작업을 확인한다.

명령어 형식	동작 내용	추가 설명
>> + string "+ *box*"	문자열 string을 현재 BTS에 들어갈 위치를 찾아 leaf 위치에 이것을 추가한다.	이미 Key값이 Tree에 있는 경우에는 변함이 없다.
>> - string "- soju"	제거할 data가 leaf에 있다면 바로 지운다. 만일 그것이 내부 노드라면 그것을 지운 다음 오른쪽 부트리의 최소값으로 대치한다. 만일 오른쪽 부트리가 없으면 왼쪽 부트리의 최대값으로 교체하고 leaf에 도달할 때까지 계속한다.	해당되는 Key가 없는 경우에는 무시한다. 따라서 BTS는 그대로 유지된다.
>> depth k	depth k에 있는 노드를 찾아 사전식 순서로 모두 출력한다. 없을 경우에는 "NO"를 출력.	root의 depth는 1이며 depth k노드의 자식 노드의 depth는 $k+1$.
>> leaf	BTS의 leaf를 왼쪽부터 사전식으로 모두 출력 root만 있을 경우에는 root를 출력	항상 한 개 이상 존재함.

[입출력] 입력의 첫 줄에는 명령어의 개수 N $(5 \le N \le 100)$이 주어진다. 그리고 이어지는 N개의 각 줄에 한 개의 명령어가 주어진다. 여러분은 명령어 "depth k"나 "leaf"에 대하여 해당되는 원소를 사전식 순서대로 한 줄에 모두 출력해야 한다.

[예제**]**

입력 **stdin**	출력 **stdout**
13 // 명령어 개수 + phone + banana - cola // 없으면 무시함 + chip **leaf** + pizza + soccer - phone **depth 3** + machine **depth 2** - pizza **leaf**	**banana phone// leaf**의 결과 **soccer// depth 3** **banana pizza //depth 2** **banana soccer // leaf**

[제한조건**]** 프로그램의 이름은 BTS.{py,c,cpp,java}이다. 제출 횟수는 최대 15번이며 허용 시간은 데이터 당 제한 시간은 1초, 허용가능 코드의 최대 크기는 3000 bytes 이다. 문제 풀이 마감시간은 **12월9일(금요일) 24:00** 이다. 제출한 프로그램에 대한 풀이(방법과 코드설명)를 작성하여 2022년 12월10일 24:00까지 NESPA "설명게시판"에 제출해야 한다. 제출한 프로그램 풀이과정은 마감이 지나면 공개된다. 제시간에 제출하지 못한 학생은 1 day 유예시간(마감 12월 10일)이 주어진다. 유예시간에는 10%의 감점이 적용된다.제시간에 제출하지 못한 학생은 1일 유예시간이 주어진다. 유예시간에는 10%의 감점이 적용된다.