

# BST, not BTS

[문제] 이진 탐색나무(Binary Search Tree)를 만들어 string 원소를 추가(insert), 삭제하는 작업을 한다. 이 작업 도중에 관리하고 있는 BST의 depth  $k$ 에 있는 node 또는 전체 leaf node를 찾아서 왼쪽부터 오른쪽 순으로 (크기 순서대로) scan하면서 모두 출력해야 한다.

명령어 형식	동작 내용	설명
>> + string "+ box"	입력 문자열 string을 현재 BST의 올바른 leaf 위치에 삽입한다.	이미 Key값이 들어있는 경우에는 변함이 없다.
>> - string "- soju"	제거할 data가 leaf에 있다면 바로 지운다. 만일 그것이 내부 노드라면 그것을 지운 다음 왼쪽 부트리의 최대값으로 대체한다. 만일 왼쪽 부트리가 없으면 오른쪽 부트리의 최소값으로 교체하고 leaf가 될 때까지 계속한다.	해당되는 Key가 없는 경우에는 무시한다. 따라서 BST는 그대로 유지된다.
>> depth $k$	depth $k$ 에 있는 노드를 찾아 왼쪽부터 출력한다. 없을 경우에는 "NO"를 출력.	root의 depth는 1이다. depth $k$ 인 노드의 자식 노드의 depth는 $k+1$ .
>> leaf	BST의 leaf를 왼쪽부터 모두 출력 root만 있을 경우에는 root를 출력	한 개 이상 존재함.

[입출력] 입력 파일 **bst.inp** 의 첫 줄에는 명령어의 개수  $N$  ( $5 \leq N \leq 50$ )이 주어진다. 그리고 이어지는  $N$ 개의 각 줄에는 명령어가 하나씩 주어진다. 여러분은 2 명령어 "depth  $k$ "이나 "leaf"에 대하여 해당되는 원소를 트리의 왼쪽에서 오른쪽 순대로 순서대로 출력해야 한다.

[보기]

bst.inp	bst.out
13 // 명령어 개수 + phone + banana - cola // 없으면 무시함 + chip <b>leaf</b> + pizza + soccer - phone <b>depth 3</b> + machine <b>depth 2</b> - pizza <b>leaf</b>	chip // leaf의 결과 soccer // depth 3 banana pizza //depth 2 banana soccer // leaf

[제한조건] 프로그램의 이름은 bst.{c, cpp, java, py} 이다. 이번 과제의 마감은 11월 24일(토요일) 저녁 10시이며 제출은 11월 20일(화요일)부터 가능하다. 제출 횟수는 최대 15번이며 허용 시간은 1초이다, 프로그램에는 채점자가 알아 볼 수 있도록 간결하면서도 적절한 설명문(comment)을 적어놓아야 한다. 이번 과제물은 기말고사에서 다시 활용할 예정이므로 잘 이해하고 관리해야 한다.