



BTS machine

[문제] 자료의 탐색 뿐만 아니라 새로운 자료의 삽입, 기존 자료의 삭제 등을 지원하는 dynamic 탐색 이진 트리 BTS(Binary Tree for Searching) 기계를 구현해보자. 이 자료구조에는 키가 string 데이터를 추가(insert), 삭제 그리고 몇 가지 추가 작업(operation)을 할 수 있다. 예를 들면 작업 중 관리되고 있는 BTS의 depth k 에 있는 node 또는 leaf node를 모두 찾아 왼쪽부터 오른쪽 순으로, 사전식 순서대로 모두 출력할 수 있어야 한다. 이 자료구조에 적용할 수 있는 동작은 다음과 같다.

명령어 형식	동작(operation)의 내용	추가 설명
>> + string "+ box"	문자열 string을 현재 BTS에 들어갈 위치를 찾아 leaf 위치에 이것을 추가한다.	이미 Key값이 Tree에 있는 경우에는 BTS의 구조에는 변함이 없다.
>> - string "- soju"	제거할 data가 leaf에 있다면 바로 지운다. 만일 그것이 내부 노드라면 그것을 지운 다음 왼쪽 부트리(left subtree)의 최대값으로 대체한다. 만일 왼쪽 부트리가 <u>없으면</u> 오른쪽 부트리의 최소값으로 교체한다. 이 작업은 지울 대상 노드가 leaf가 될 때까지 반복한다. 즉 중간노드를 지우는 작업은 각 subtree에서 재귀적으로 적용된다.	해당되는 Key가 대상 tree에 없는 경우에는 이 동작은 무시된다. 따라서 BTS는 그대로 유지된다.
>> depth k	depth k 에 있는 노드를 찾아 사전식 순서로 모두 출력한다. 없을 경우에는 "NO"를 출력.	root의 depth는 1이며 depth k 노드의 자식 노드의 depth는 $k+1$.
>> leaf	BTS의 leaf를 모두 찾아서 <u>사전식 순서</u> 로 모두 출력한다. 만일 단일 노드 root만 있을 경우에는 이 root를 출력해야 한다.	단 대상 트리에는 항상 한 개 이상 존재함. 적어도 root는 존재한다.
>> quit	입력의 "끝"을 의미하며 이것으로 종료한다.	더 이상 "명령"은 없다.

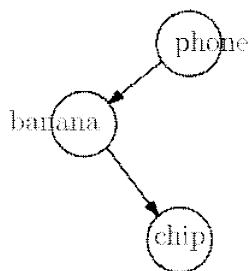
[입출력] 입력과 출력은 이전 과제와 마찬가지로 표준(standard) stdin, stdout이다. 입력 파일 stdin의 첫

1) Sorry. In this class, BTS (Binary Tree for Searching) does not mean the singer group, "방탄소년단".

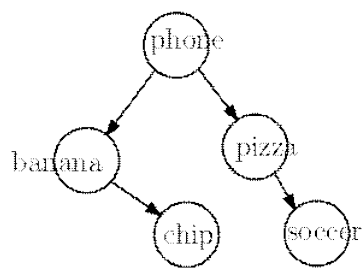
줄부터 위에 제시한 명령어가 한 줄에 하나씩 나오며 그 마지막은 명령어 "**quit**"으로 표시된다. 이 **quit** 명령어를 받으면 작업을 종료한다. 여러분은 2 명령어 "**depth k**", "**leaf**"에 대하여 해당되는 원소를 사전식(lexicographic) 순서대로 한 줄에 공백을 두고 모두 출력해야 한다. 명령어의 최대 갯수는 2개 이상 100개 이하이다.

[보기]

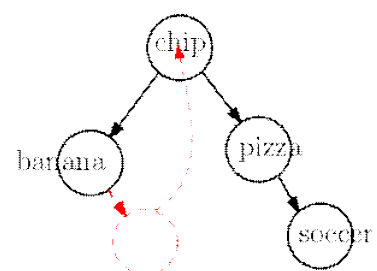
stdin 입력	stdout 출력
+ phone	chip // leaf의 결과
+ banana	soccer // depth 3
- cola // 없으므로 무시함	banana pizza //depth 2
+ chip	banana soccer // leaf
leaf	
+ pizza	
+ soccer	
- phone	
depth 3	
+ machine	
depth 2	
- pizza	
leaf	
quit // 입력 끝....	



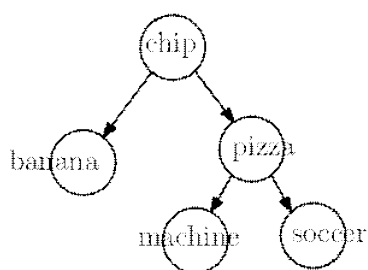
step) 1-5



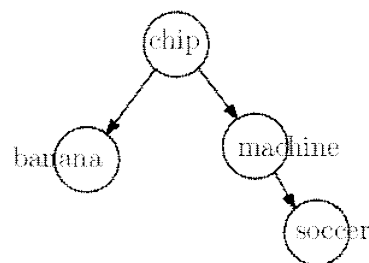
step) 6-7



step) 8-9, : - phone



step) 10



step) 11 : - pizza

[제한조건] 프로그램의 이름은 **BTS.{c, cpp, py}** 이다. 제출 횟수는 최대 10번이며 허용 시간은 1초이다. 사용할 수 있는 토큰의 최대 갯수는 **700**이다.

이번 과제물은 기말고사(조환규 교수 분반의 경우)에서 활용될 수 있으므로 잘 이해하고 있어야 한다. 즉 기말고사 코딩 문제에서 입력은 동일하다. 첫 10번째 만점자 중에서 가장 “우아한” 코드를 작성한 3명에게 보너스 추가 5점을 부여한다. 따라서 가능하면 빨리 제출해야 한다. 단 선정된 학생은 성실하게 자신의 코드를 “평가 게시판”에서 review 해야 한다.