

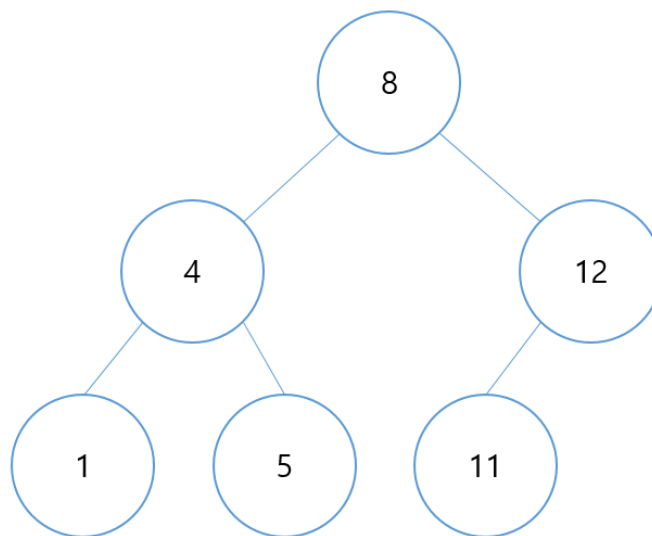
Week 11

- 주의사항: 부정행위금지 (채점 서버 외 인터넷 사용 금지), STL 사용금지(string는 사용 가능)
- 표준 입출력 사용을 권장(C는 scanf / printf, C++은 cin / cout)

문제 1

비룡이는 자료구조 수업에서 데이터의 삽입과 탐색이 빈번하게 발생할 경우, 이진탐색트리(binary search tree)를 사용하면 효율적이라는 것을 배웠다. 비룡이는 다음의 과정을 통해 이진탐색트리가 얼마나 효율적인지 확인해보려 한다.

1. 이진탐색트리에 삽입 연산을 수행할 때, 키(key) 간의 비교 연산을 몇 번 수행하는지 집계한다.
2. 이진탐색트리에서 탐색 연산을 수행할 때, 키 간의 비교 연산을 몇 번 수행하는지 집계한다.



예를 들어, 위 그림과 같이 정수를 저장하는 이진탐색트리에 대해 삽입 또는 탐색 연산을 수행한다고 하자.

- 예시 1) 정수 7을 삽입한다면, 노드 8, 4, 5와 차례로 비교하여 3번의 비교 연산을 사용한다.
- 예시 2) 정수 14를 탐색한다면, 노드 8, 12와 차례로 비교하여 2번의 비교 연산을 사용한다.
- 예시 3) 정수 12를 탐색한다면, 노드 8, 12와 차례로 비교하여 2번의 비교 연산을 사용한다.

비룡이를 도와, 다음의 명령어를 처리하는 이진탐색트리를 구현하시오.

- **insert x** : 이진탐색트리에 정수 x ($1 \leq x \leq 100,000$)를 삽입하고, 그 과정에서 필요한 비교 연산의 횟수를 출력한다. 단, x 는 이진탐색트리 내에 존재하지 않는 수이다.

- **search x** : 이진탐색트리에서 노드 x ($1 \leq x \leq 100,000$)를 찾는 과정에서 필요한 비교 연산의 횟수를 출력한다. 단, 이진탐색trie가 비어 있다면 "empty"를 출력한다. x 는 이진탐색trie 내에 존재하지 않을 수도 있다.
- **parent x** : 이진탐색trie에서 노드 x ($1 \leq x \leq 100,000$) 를 찾아서, x 의 부모의 키 값을 출력한다. 만약 노드 x 가 이진탐색trie에 존재하지 않다면 -1 을, x 의 부모가 존재하지 않다면 -2 를 출력한다.
- **child x** : 이진탐색trie에서 노드 x ($1 \leq x \leq 100,000$)를 찾아서, x 의 왼쪽 자식의 키 값을 출력한다. 만약 노드 x 가 이진탐색trie에 존재하지 않다면 -1 을, x 의 왼쪽 자식이 존재하지 않다면 -2 를 출력한다.

입력

이 문제는 여러 개의 테스트 케이스로 이루어져 있다.

첫 번째 줄에 테스트 케이스의 수 T ($1 \leq T \leq 10$)가 주어진다.

각 테스트 케이스는 다음과 같이 이루어져 있다.

- 첫 번째 줄에 명령어의 개수 N ($1 \leq N \leq 10,000$)이 주어진다.
- N 개의 줄에 걸쳐 명령어가 주어진다.

출력

각 테스트 케이스에서, 출력이 필요한 명령이 주어질 때마다 그 결과를 한 줄에 하나씩 출력한다.

예제 입출력

예제 입력	예제 출력
1	0
11	1
insert 8	2
insert 4	2
insert 1	1
insert 5	2
insert 12	3
insert 11	2
insert 7	2
search 12	-2
search 14	4
parent 8	
child 8	