[SWEA]1231

0316

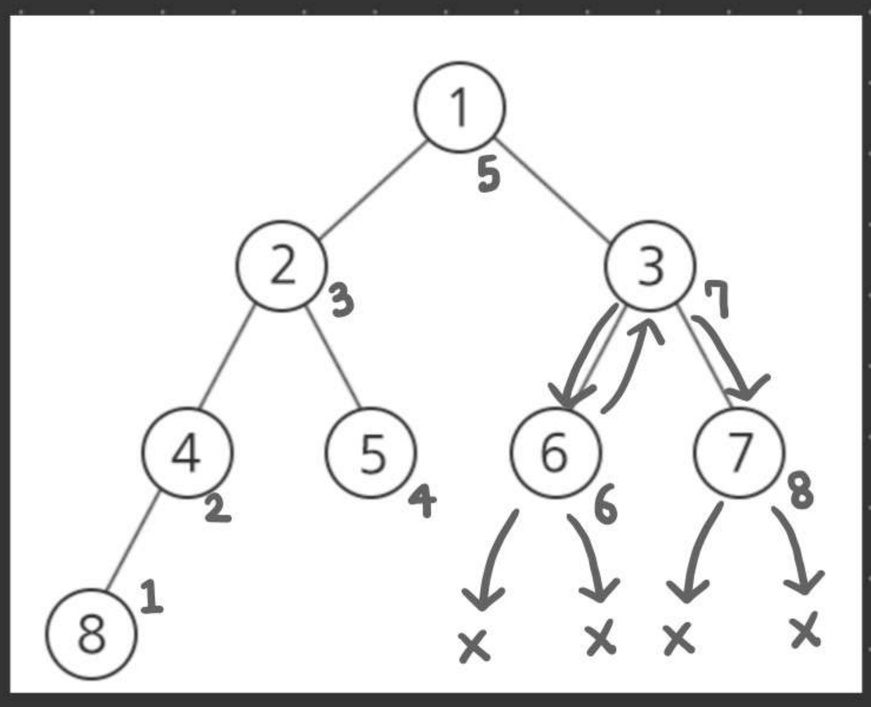
중간순회

다음은 특정 단어(또는 문장)를 트리 형태로 구성한 것으로, in-order 형식으로 순회하여 각 노드를 읽으면 원래 단어를 알 수 있다고 한다.
위 트리를 in-order 형식으로 순회할 경우 SOFTWARE 라는 단어를 읽을 수 있다.

[제약 사항]
총 10개의 테스트 케이스가 주어진다.총 노드의 개수는 100개를 넘어가지 않는다. 트리는 완전 이진 트리 형식으로 주어지며, 노드당 하나의 알파벳만 저장할 수 있다. 노드가 주어지는 순서는 아래 그림과 같은 숫자 번호대로 주어진다.

[입력]
각 테스트 케이스의 첫 줄에는 각 케이스의 트리가 갖는 정점의 총 수 N(1≤N≤100)이 주어진다. 그 다음 N줄에 걸쳐 각각의 정점 정보가 주어진다.
해당 정점에 대한 정보는 해당 정점의 알파벳, 해당 정점의 왼쪽 자식, 오른쪽 자식의 정점번호가 차례대로 주어진다.
정점번호는 1부터 N까지의 정수로 구분된다. 입력에서 정점 번호를 매기는 규칙은 위와 같으며, 루트 정점의 번호는 반드시 1이다.
입력에서 이웃한 알파벳이나 자식 정점의 번호는 모두 공백으로 구분된다.
총 10개의 테스트 케이스가 주어진다.

[출력]
#부호와 함께 테스트 케이스의 번호를 출력하고, 공백 문자 후 테스트 케이스의 답을 출력한다.



순서: 왼쪽 자식 → 부모 → 오른쪽 자식

예시) N=8
arr=[(1,W,2,3),(2,F,4,5),(3,R,6,7),
(4,O,8),(5,T),(6,A),(7,E),(8,S)]

i) 인덱스 : 정점번호
값 : 단어
) 인 리스트 생성


```
def in_order(v):
    if V ≤ N:
        in_order(v*2)
        Word[V]출력
        in_order(v*2+1)
```

ii) 인덱스 순회하여 출력하기

왼쪽자식노드 → 부모 → 오른쪽 자식노드
i*2 i i*2+1

- 재귀로 구현
  조건 : 인덱스(정점번호) ≤ 마지막 정점번호

예시

in_order(1) V=1
1) in_order(v*2) — in_order(2) V=2
14) word[1]출력
15) in_order(v*2+1)
: 반복

in_order(2) V=2
2) in_order(v*2) — in_order(4) V=4
9) Word[2]출력
10) in_order(v*2+1)
in_order(5) V=5
11) in_order(v*2)
12) word[5]출력
13) in_order(v*2+1)

in_order(4) V=4
3) in_order(v*2) — in_order(8) V=8
7) word[4]출력
8) in_order(v*2+1)
in_order(9) V=9(V>N)

in_order(8) V=8
4) in_order(v*2) — in_order(16) V=16(V>N)
5) word[8]출력
6) in_order(v*2+1) — in_order(17) V=17(V>N)