[SWEA]1232 사칙연산

0317

사칙연산으로 구성되어 있는 식은 이진 트리로 표현할 수 있다. 아래는 식 “(9/(6-4))*3”을 이진 트리로 표현한 것이다.
임의의 정점에 연산자가 있으면 해당 연산자의 왼쪽 서브 트리의 결과와 오른쪽 서브 트리의 결과를 사용해서 해당 연산자를 적용한다.

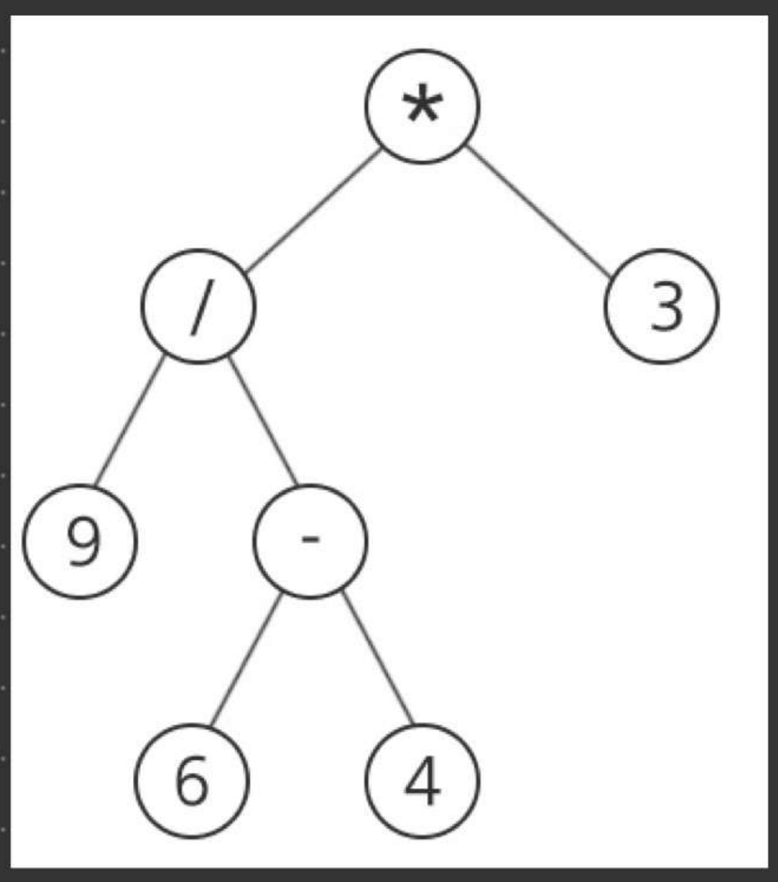
사칙연산 “+, -, *, /”와 양의 정수로만 구성된 임의의 이진트리가 주어질 때, 이를 계산한 결과를 출력하는 프로그램을 작성하라.
단, 중간 과정에서의 연산은 실수 연산으로 하되, 최종 결과값이 정수로 떨어지지 않으면 정수부만 출력한다. 위의 예에서는 최종 결과값이 13.5이므로 13을 출력하면 된다.

[제약 사항]
정점의 총 수 N은 1≤N≤1000.

[입력]
각 테스트 케이스의 첫 줄에는 각 케이스의 트리가 갖는 정점의 총 수 N(1≤N≤1000)이 주어진다. 그 다음 N줄에 걸쳐 각각의 정점 정보가 주어진다.
정점이 단순한 수이면 정점번호와 해당 양의 정수가 주어지고, 정점이 연산자이면 정점번호, 연산자, 해당 정점의 왼쪽 자식, 오른쪽 자식의 정점번호가 차례대로 주어진다.정점번호는 1부터 N까지의 정수로 구분된다.
입력에서 정점 번호를 매기는 특별한 규칙은 없으나, 루트 정점의 번호는 반드시 1이다. 입력에서 이웃한 수나 연산자는 모두 공백으로 구분된다.

위의 예시에서, 숫자 4가 7번 정점에 해당하면 “7 4”으로 주어지고, 연산자 ‘/’가 2번 정점에 해당하면 두 자식이 각각 숫자 9인 4번 정점과 연산자 ‘-’인 5번 정점이므로 “2 / 4 5”로 주어진다. 총 10개의 테스트 케이스가 주어진다.

[출력]
#부호와 함께 테스트 케이스의 번호를 출력하고, 공백 문자 후 테스트 케이스에 대한 답을 출력한다. 답은 항상 정수값으로 기록한다.

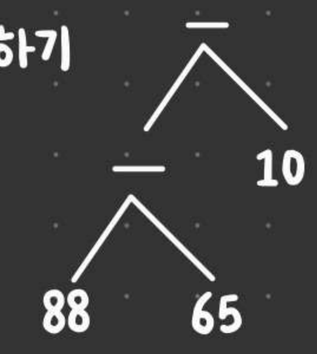


예시) N=5
arr = [(1,−,2,3),(2,−,4,5),(3,10),
(4,88),(5,65)]

왼쪽 | 0 | 2 | 4 | | | |

오른쪽 | 0 | 3 | 5 | | | |

i) 트리 후위 순회하기



tree = [88,65,−,10,−]
88-65=23
23-10

ii) Stack으로 계산하기
↳ 리스트를 순회한다 (tree)

88 — 1) 88은 피연산자이므로 Stack에 추가

88 65 — 2) 65는 피연산자이므로 Stack에 추가

23 — 3) −는 연산자이므로 계산하기위한 피연산자 필요
↳ Stack에서 피연산자 빼오기!
∴ 88-65=23 ⟶ Stack에 추가

⋮ 반복

[SWEA]5174 Subtree

0317

트리의 일부를 서브 트리라고 한다. 주어진 이진 트리에서 노드 N을 루트로 하는 서브 트리에 속한 노드의 개수를 알아내는 프로그램을 만드시오.

주어지는 트리는 부모와 자식 노드 번호 사이에 특별한 규칙이 없고, 부모가 없는 노드가 전체의 루트 노드가 된다.자식 노드가 0인 경우는 노드가 자식이 없는 경우이다.

[입력]
첫 줄에 테스트케이스의 수 T가 주어진다. 1<=T<=50
다음 줄부터 테스트 케이스의 별로 첫 줄에 간선의 개수 E와 N이 주어지고, 다음 줄에 E개의 부모 자식 노드 번호 쌍이 주어진다.
노드 번호는 1번부터 E+1번까지 존재한다. 1<=E<=1000, 1<=N<=E+1

[출력]
각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 답을 출력한다.

예시) E = 5
N = 1
arr = [(2,1),(2,5),(1,6),(5,3),(6,4)]

i) 정점의 개수 : V = E+1

ii) left부터 채운다

부모번호:	1	2	3	4	5	6
자식(left) :	0	6	1	0	0	3
자식(right):	0	0	5	0	0	0

반복문

iii) N을 Root로 하는 트리 순회 (재귀)
정점 방문할 때마다 count!
반복조건 : 자식이 없으면 반복 종료
(Sigle node)

[SWEA]5176 이진탐색

중위탐색

1부터 N까지의 자연수를 이진 탐색 트리에 저장하려고 한다. 이진 탐색 트리는 어떤 경우에도 저장된 값이 왼쪽 서브트리의 루트 <현재 노드 <오른쪽 서브 트리의 루트인 규칙을 만족한다.추가나 삭제가 없는 경우에는, 완전 이진 트리가 되도록 만들면 효율적인 이진 탐색 트리를 만들수 있다.완전 이진 트리의 노드 번호는 루트를 1번으로 하고 아래로 내려가면서 왼쪽에서 오른쪽 순으로 증가한다.

N이 주어졌을 때 완전 이진 트리로 만든 이진 탐색 트리의 루트에 저장된 값과, N/2번 노드(N이 홀수인 경우 소수점 버림)에 저장된 값을 출력하는 프로그램을 만드시오.

[입력]
첫 줄에 테스트케이스의 수 T가 주어진다. 1<=T<=50
다음 줄부터 테스트 케이스의 별로 N이 주어진다. 1<=N<=1000

[출력]
각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 답을 출력한다.

예시) N = 6 ⟶ 정점의 값 : [1, 2, 3, 4, 5, 6]

완전이진트리이기 때문에

∴ 중위우선탐색을 통해 정점번호의 순서를 찾고 값을 순서대로 대입한다.

ex) 정점의 값:

정점번호 :	4	2	5	1	6	3

(순회순서로나열)

[SWEA]5177 이진힙

0317

이진 최소힙은 다음과 같은 특징을 가진다.
- 항상 완전 이진 트리를 유지하기 위해 마지막 노드 뒤에 새 노드를 추가한다.
- 부모 노드의 값<자식 노드의 값을 유지한다. 새로 추가된 노드의 값이 조건에 맞지 않는 경우, 조건을 만족할 때까지 부모 노드와 값을 바꾼다.
- 노드 번호는 루트가 1번, 왼쪽에서 오른쪽으로, 더 이상 오른쪽이 없는 경우 다음 줄로 1씩 증가한다.

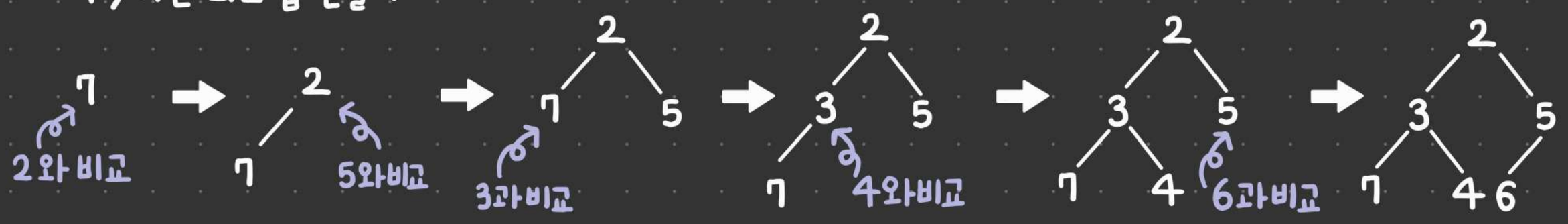
1000000이하인 N개의 서로 다른 자연수가 주어지면 입력 순서대로 이진 최소힙에 저장하고, 마지막 노드의 조상 노드에 저장된 정수의 합을 알아내는 프로그램을 작성하시오.

[입력]
첫 줄에 테스트케이스의 수 T가 주어진다. 1<=T<=50
다음 줄부터 테스트 케이스의 별로 N이 주어지고, 다음 줄에 1000000이하인 서로 다른 N개의 자연수가 주어진다. 5<=N<=500

[출력]
각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 답을 출력한다.

예시) N = 6 arr = [7, 2, 5, 3, 4, 6]

i) 이진 최소 힙 만들기



ii) 마지막 노드의 조상노드에 저장된 값을 더한다

N번노드 N//2 ⟶ 1일 때 까지 반복해서 더한다

ROOT

[SWEA]5178 노드의 합

완전 이진 트리의 리프 노드에 1000이하의 자연수가 저장되어 있고, 리프 노드를 제외한 노드에는 자식 노드에 저장된 값의 합이 들어있다고 한다. N개의 노드를 갖는 완전 이진 트리의 노드 번호는 루트가 1번이 되며, 같은 단계에서는 왼쪽에서 오른쪽으로 증가, 단계가 꽉 차면 다음단계의 왼쪽부터 시작된다.완전 이진 트리의 특성상 1번부터 N번까지 빠지는 노드 번호는 없다. 리프 노드의 번호와 저장된 값이 주어지면 나머지 노드에 자식 노드 값의 합을 저장한 다음, 지정한 노드 번호에 저장된 값을 출력하는 프로그램을 작성 하시오.

[입력]
첫 줄에 테스트케이스의 수 T가 주어진다. 1<=T<=50
다음 줄부터 테스트 케이스의 별로 노드의 개수 N과 리프 노드의 개수 M, 값을 출력할 노드 번호 L이 주어지고, 다음 줄부터 M개의 줄에 걸쳐 리프 노드 번호와 1000이하의 자연수가 주어진다.

[출력]
각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 답을 출력한다.

예시) N=5 M=3 L=2
arr = [(4,1), (5,2), (3,3)]

즉, leaf를 tree에 기입하고 거꾸로 부모의 값을 채운다

i) 부모정점번호 = 자식정점번호//2

ii) 부모정점의 값 = 자식정점의 합