

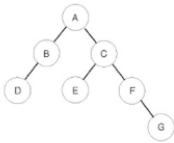
트리 순회



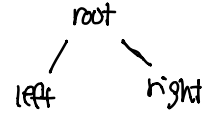
시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	15352	9419	7201	62.946%

문제

이진 트리를 입력받아 **전위 순회(preorder traversal)**, **중위 순회(inorder traversal)**, **후위 순회(postorder traversal)**한 결과를 출력하는 프로그램을 작성하시오.



preorder : A B D C E F G
 inorder : D B A E C F G
 postorder : D B E G F C A



예를 들어 위와 같은 이진 트리가 입력되면,

- 전위 순회한 결과 : ABCDEFG // (루트) (왼쪽 자식) (오른쪽 자식)
- 중위 순회한 결과 : DBAECFG // (왼쪽 자식) (루트) (오른쪽 자식)
- 후위 순회한 결과 : DBEGFCA // (왼쪽 자식) (오른쪽 자식) (루트)

가 된다.

입력

첫째 줄에는 **이진 트리의 노드의 개수** N ($1 \leq N \leq 26$)이 주어진다. 둘째 줄부터 N 개의 줄에 걸쳐 각 노드와 그의 왼쪽 자식 노드, 오른쪽 자식 노드가 주어진다. 노드의 이름은 A 부터 차례대로 영문자 대문자로 매겨지며, 항상 A가 루트 노드가 된다. 자식 노드가 없는 경우에는 .으로 표현된다.

출력

첫째 줄에 전위 순회, 둘째 줄에 중위 순회, 셋째 줄에 후위 순회한 결과를 출력한다. 각 줄에 N 개의 알파벳을 공백 없이 출력하면 된다.

예제 입력 1 복사

```

7
A B C
B D .
C E F
E . .
F . G
D . .
G . .
  
```

예제 출력 1 복사

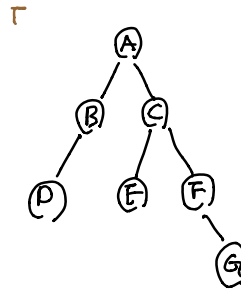
```

ABDCEFG
DBAECFG
DBEGFCA
  
```

↳ node class 생성

↳ left, right 추가하기

⊕ → dictionary 활용



다음과 같이 tree 완성

→ Traversal 에 대해서는 각

↳ pre Order
 in Order
 post Order

left child
 data
 right child

순서를 변경

트리의 높이와 너비



시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	7381	2190	1499	27.505%

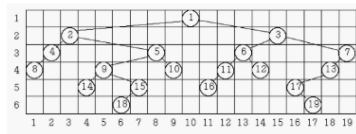
문제

이진트리를 다음의 규칙에 따라 행과 열에 번호가 붙어있는 격자 모양의 틀 속에 그리려고 한다. 이때 다음의 규칙에 따라 그리려고 한다.

- 이진트리에서 **같은 레벨(level)에 있는 노드는 같은 행에** 위치한다.
- 한 열에는 한 노드만** 존재한다.
- 왼쪽의 노드의 왼쪽 부분리(left subtree)에 있는 **노드들은 해당 노드보다 왼쪽의 열에** 위치하고, **오른쪽 부분리(right subtree)에 있는 노드들은 해당 노드보다 오른쪽의 열에** 위치한다.
- 노드가 배치된 **가장 왼쪽 열과 오른쪽 열 사이엔 아무 노드도 없어야**하는 열은 없다. **→ 이 부분이 중요하다**

이와 같은 규칙에 따라 이진트리를 그릴 때 각 레벨의 너비는 그 레벨에 할당된 노드 중 가장 오른쪽에 위치한 노드의 열 번호에서 가장 왼쪽에 위치한 노드의 열 번호를 뺀 값 더하기 1로 정의된다. 트리의 레벨은 가장 왼쪽에 있는 루트 노드가 1이고 아래로 1씩 증가한다.

아래 그림은 어떤 이진트리를 위의 규칙에 따라 그려 본 것이다. 첫 번째 레벨의 너비는 1, 두 번째 레벨의 너비는 13, 3번째, 4번째 레벨의 너비는 각각 18이고, 5번째 레벨의 너비는 13이며, 그리고 6번째 레벨의 너비는 12이다.



우리는 주어진 이진트리를 위의 규칙에 따라 그림 때에 **너비가 가장 넓은 레벨과 그 레벨의 너비를 계산**하려고 한다. 위의 그림의 예에서 너비가 가장 넓은 레벨은 3번째와 4번째로 그 너비는 18이다. 너비가 가장 넓은 레벨이 두 개 이상 있을 때는 번호가 작은 레벨을 답으로 한다. 그러므로 이 예에 대한 답은 3이고, 너비는 18이다.

임의의 이진트리가 입력으로 주어질 때 너비가 가장 넓은 레벨과 그 레벨의 너비를 출력하는 프로그램을 작성하시오

입력

첫째 줄에 노드의 개수를 나타내는 정수 N ($1 \leq N \leq 10,000$)이 주어진다. 다음 N 개의 줄에는 각 줄마다 노드 번호와 해당 노드의 왼쪽 자식 노드와 오른쪽 자식 노드의 번호가 순서대로 주어진다. 노드들의 번호는 1부터 N 까지이며, 자식이 없는 경우에는 **자식 노드의 번호에 -1**이 주어진다.

출력

첫째 줄에 너비가 가장 넓은 레벨과 그 레벨의 너비를 순서대로 출력한다. 너비가 가장 넓은 레벨이 두 개 이상 있을 때에는 번호가 작은 레벨을 출력한다.

예제 입력 1 복사

```
19
1 2 3
2 4 5
3 6 7
4 8 -1
5 9 10
6 11 12
7 13 -1
8 -1 -1
9 14 15
10 -1 -1
11 16 -1
12 -1 -1
13 17 -1
14 -1 -1
15 18 -1
16 -1 -1
17 -1 19
18 -1 -1
19 -1 -1
```

예제 출력 1 복사

```
3 18
```

→ **중위순회** 통해
숫자 부여하기

※

inorder 과정 속에서

각 level 별 최소 xpos

최대 xpos

7 저장



마지막에, 각 level 별 「max_xpos - min_xpos」

↳ 최대 (max_xpos - min_xpos) 찾기

⊕ 이때의 level

최소 힙

☆

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초 (추가 시간 없음)	128 MB	1,4133	5719	4398	45.927%

문제

널리 잘 알려진 자료구조 중 최소 힙이라는 것이 있다. 최소 힙을 이용하여 다음과 같은 연산을 지원하는 프로그램을 작성하시오.

- 배열에 자연수 x 를 넣는다.
- 배열에서 가장 작은 값을 출력하고, 그 값을 배열에서 제거한다.

프로그램은 처음에 비어있는 배열에서 시작하게 된다.

입력

첫째 줄에 연산의 개수 N ($1 \leq N \leq 100,000$)이 주어진다. 다음 N 개의 줄에는 연산에 대한 정보를 나타내는 정수 x 가 주어진다. 만약 x 가 자연수라면 배열에 x 라는 값을 넣는(추가하는) 연산이고, x 가 0이라면 배열에서 가장 작은 값을 출력하고 그 값을 배열에서 제거하는 경우이다. 입력되는 자연수는 2^{31} 보다 작다.

출력

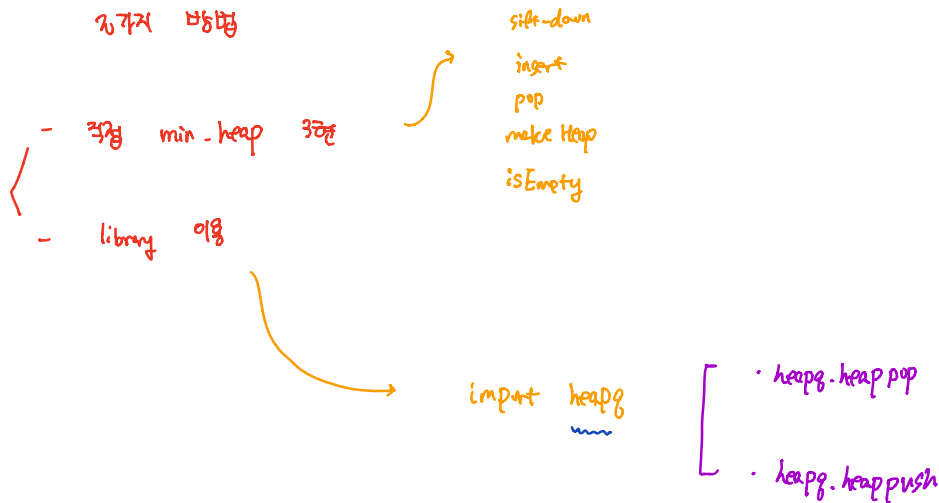
입력에서 0이 주어진 회수만큼 답을 출력한다. 만약 배열이 비어 있는 경우인데 가장 작은 값을 출력하라고 한 경우에는 0을 출력하면 된다.

예제 입력 1 복사

```
9
0
12345678
1
2
0
0
0
0
0
32
```

예제 출력 1 복사

```
0
1
2
12345678
0
```





1715년

재출

맞은 사람

숫고당

재제정/수정

재정 원형

내 소스

강의 >

질문 검색

카드 정렬하기



시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	4416	1541	1327	38.699%

문제

정렬된 두 묶음의 숫자 카드가 있다고 하자. 각 묶음의 카드의 수를 A, B라 하면 보통 두 묶음을 합쳐서 하나로 만드는 데에는 A+B 번의 비교를 해야 한다. 이를테면, 20장의 숫자 카드 묶음과 30장의 숫자 카드 묶음을 합치려면 50번의 비교가 필요하다.

매우 많은 숫자 카드 묶음이 책상 위에 놓여 있다. 이들을 두 묶음씩 골라 서로 합쳐나간다면, 고르는 순서에 따라서 비교 횟수가 매우 달라진다. 예를 들어 10장, 20장, 40장의 묶음이 있다면 10장과 20장을 합친 뒤, 합친 30장 묶음과 40장을 합친다면 $(10+20)+(30+40) = 100$ 번의 비교가 필요하다. 그러나 10장과 40장을 합친 뒤, 합친 50장 묶음과 20장을 합친다면 $(10+40)+(50+20) = 120$ 번의 비교가 필요하므로 덜 효율적인 방법이다.

N개의 숫자 카드 묶음의 각각의 크기가 주어질 때, 최소한 몇 번의 비교가 필요한지를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N이 주어진다. ($1 \leq N \leq 100,000$) 이어서 N개의 줄에 걸쳐 숫자 카드 묶음의 각각의 크기가 주어진다.

출력

첫째 줄에 최소 비교 횟수를 출력한다. (21억 이하)

예제 입력 1 복사

```
3
10
20
40
```

예제 출력 1 복사

100

20 + 40 + 40

$$\rightarrow 10 \quad 20 \quad 40 = 10+20 + (10+20)+40 = 100$$

10 20 40

$$\rightarrow 10 \quad 40 \quad 20 = 10+40 + (10+40)+20 = 120$$

$$\rightarrow 20 \quad 40 \quad 10 = \frac{20+40}{60} + \frac{20+40}{60} + 10 = 130$$

30
10 20 30 40

→ 26명 캐서 더해준다.

$$f(n) = f(n-1) + f(n-2) + f(n) \quad 20 \quad 40 \quad \frac{20+40+10}{60}$$

$$a_n = \sum_{i=1}^{n-1} a_i$$

10 20 30
10 20 30 40

$$\Rightarrow \text{Sum} \Rightarrow \frac{n-1 a_0 + n-1 a_1 + \dots + a_{n-1}}{n}$$

↳ x general solution

⇒ consider local minimum

↓
X right solution

greedy

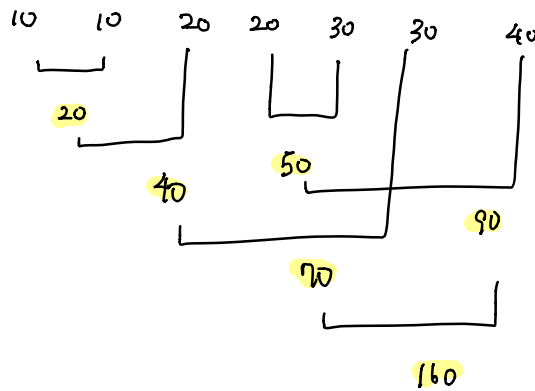
ex) 10 10 20 20 30 30 40

by the wrong formula

$$\rightarrow 10 \times 6 + 10 \times 6 + 20 \times 5 + 20 \times 4 + 30 \times 3 + 30 \times 2 + 40 \times 1$$

$$= 60 + 60 + 100 + 80 + 90 + 60 + 40 = 530$$

but, right answer



$$20 + 40 + 50 + 70 + 90 + 160$$

↑ 30

• 중간 결과를 배열에 넣어서 다시 정렬 필요

→

- data = pop() + pop()
- sum += data
- push(data)

문제집



시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	128 MB	7455	3387	2572	46.713%

문제

민오는 1번부터 N번까지 총 N개의 문제로 되어 있는 문제집을 풀려고 한다. 문제는 난이도 순서로 출제되어 있다. 즉 1번 문제가 가장 쉬운 문제이고 N번 문제가 가장 어려운 문제가 된다.

어떤 문제부터 풀까 고민하면서 문제를 풀어보던 민오는, 몇몇 문제들 사이에는 '먼저 푸는 것이 좋은 문제'가 있다는 것을 알게 되었다. 예를 들어 1번 문제를 풀고 나면 4번 문제가 쉽게 풀린다거나 하는 식이다. 민오는 다음의 세 가지 조건에 따라 문제를 풀 순서를 정하기로 하였다.

1. N개의 문제는 모두 풀어야 한다.
2. 먼저 푸는 것이 좋은 문제가 있는 문제는, 먼저 푸는 것이 좋은 문제를 반드시 먼저 풀어야 한다.
3. 가능하면 쉬운 문제부터 풀어야 한다.

예를 들어서 네 개의 문제가 있다고 하자. 4번 문제는 2번 문제보다 먼저 푸는 것이 좋고, 3번 문제는 1번 문제보다 먼저 푸는 것이 좋다고 하자. 만일 4-3-2-1의 순서로 문제를 풀게 되면 조건 1과 조건 2를 만족한다. 하지만 조건 3을 만족하지 않는다. 4보다 3을 충분히 먼저 풀 수 있기 때문이다. 따라서 조건 3을 만족하는 문제를 풀 순서는 3-1-4-2가 된다.

문제의 개수와 먼저 푸는 것이 좋은 문제에 대한 정보가 주어졌을 때, 주어진 조건을 만족하면서 민오가 풀 문제의 순서를 결정해 주는 프로그램을 작성하시오.

입력

첫째 줄에 문제의 수 N ($1 \leq N \leq 32,000$)과 먼저 푸는 것이 좋은 문제에 대한 정보의 개수 M ($1 \leq M \leq 100,000$)이 주어진다. 둘째 줄부터 M개의 줄에 걸쳐 두 정수의 순서 쌍 A, B가 빈칸을 사이에 두고 주어진다. 이는 A번 문제는 B번 문제보다 먼저 푸는 것이 좋다는 의미이다.

항상 문제를 모두 풀 수 있는 경우만 입력으로 주어진다.

출력

첫째 줄에 문제 번호를 나타내는 1 이상 N 이하의 정수들을 민오가 풀어야 하는 순서대로 빈칸을 사이에 두고 출력한다.

예제 입력 1 복사

```
4 2
4 2
3 1
```

예제 출력 1 복사

```
3 1 4 2
```

