

## 철도 2 (railroad2)

IOI 나라는  $N$ 개의 도시와 도시들을 잇는  $N - 1$ 개의 양방향 철도로 이루어져 있으며, 임의의 서로 다른 두 도시를 철도만을 사용하여 오갈 수 있다. 즉, IOI 나라의 철도망은 트리 구조를 이룬다. 도시에는 각각 0 이상  $N - 1$  이하의 서로 다른 번호가 붙어 있고, 철도에도 각각 0 이상  $N - 2$  이하의 서로 다른 번호가 붙어 있다. 모든  $0 \leq i \leq N - 2$ 에 대하여  $i$ 번 철도는  $U[i]$ 번 도시와  $V[i]$ 번 도시를 양방향으로 연결하며, 철도의 길이는  $W[i]$ 이다.

IOI 나라의 어떤 도시에서 출발하더라도 다른 도시로 직통 열차를 타고 바로 이동할 수 있다. 즉,  $0 \leq u, v \leq N - 1, u \neq v$ 인 모든  $N(N - 1)$ 개의 순서쌍  $(u, v)$ 에 대해,  $u$ 번 도시에서 출발하여  $v$ 번 도시에 도착하는 직통 열차가 있다.  $u$ 번 도시에서 이 직통 열차를 타면  $v$ 번 도시에 도착할 때까지 내릴 수 없으며, 이 직통 열차의 소요 시간은 IOI 나라의 철도망에서  $u$ 번 도시에서 시작하여  $v$ 번 도시에서 끝나는 유일한 단순 경로 상의 철도들의 길이를 합한 것과 같다.

철도 동호인인 당신은 오랫동안 한 기차를 타면서 여유로움을 느끼는 것을 즐기기 때문에, 소요 시간이 긴 직통 열차만을 타고 다닐수록 더 큰 즐거움을 느낀다.

구체적으로, 서로 다른 두 도시  $x, y$ 에 대해서, 즐거움  $\text{joy}(x, y)$  는 다음 조건을 만족하는 최대의 양의 정수  $D$ 로 정의된다:

- $x$ 번 도시에서 시작하여 소요 시간이  $D$  이상인 직통 열차만을 타고 이동하는 것을 유한 번 반복하여,  $y$ 번 도시에 도착할 수 있다.

$0 \leq x, y \leq N - 1, x \neq y$ 를 만족하는 모든  $N(N - 1)$ 가지의 순서쌍  $(x, y)$ 에 대한  $\text{joy}(x, y)$ 의 합을 1000000007 ( $= 10^9 + 7$ ) 로 나눈 나머지를 구하는 프로그램을 작성하라.

## 함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
int travel(vector<int> U, vector<int> V, vector<int> W)
```

- 이 함수는 단 한 번만 호출된다.
- $U, V, W$ : 크기가  $N - 1$ 인 정수 배열. 모든  $i$  ( $0 \leq i \leq N - 2$ )에 대해,  $U[i]$ 번 도시와  $V[i]$ 번 도시를 잇는 길이  $W[i]$ 의 철도가 있다.
- 이 함수는  $0 \leq x, y \leq N - 1, x \neq y$ 를 만족하는 모든  $N(N - 1)$ 가지의 순서쌍  $(x, y)$ 에 대한  $\text{joy}(x, y)$ 의 합을 1000000007 ( $= 10^9 + 7$ ) 로 나눈 나머지를 반환해야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

## 제약 조건

- $2 \leq N \leq 500\,000$
- IOI 나라의 철도망은 트리 구조를 이룬다.
- 모든  $i$ 에 대해  $0 \leq U[i], V[i] \leq N - 1; U[i] \neq V[i]$  ( $0 \leq i \leq N - 2$ )
- 모든  $i$ 에 대해  $1 \leq W[i] \leq 1\,000\,000\,000$  ( $0 \leq i \leq N - 2$ )

## 부분문제

1. (3점)
  - $N \leq 50$
2. (6점)
  - $N \leq 500$
3. (19점)
  - $N \leq 2\,000$
4. (5점)
  - $N \leq 8\,000$
  - 모든  $i$ 에 대해  $U[i] = 0$  ( $0 \leq i \leq N - 2$ )
5. (7점)
  - $N \leq 8\,000$
  - 모든  $i$ 에 대해  $U[i] = i; V[i] = i + 1$  ( $0 \leq i \leq N - 2$ )
6. (15점)
  - $N \leq 8\,000$
7. (4점)
  - 모든  $i$ 에 대해  $U[i] = 0$  ( $0 \leq i \leq N - 2$ )
8. (11점)
  - 모든  $i$ 에 대해  $U[i] = i; V[i] = i + 1$  ( $0 \leq i \leq N - 2$ )
9. (30점)
  - 추가적인 제약 조건이 없다.

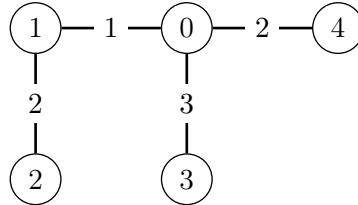
### 예제 1

$N = 5, U = [0, 1, 0, 0], V = [1, 2, 3, 4], W = [1, 2, 3, 2]$ 인 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
travel([0, 1, 0, 0], [1, 2, 3, 4], [1, 2, 3, 2])
```

아래 그림은 도시들과 철도망들의 배치를 나타낸다.



$x$ 번 도시와  $y$ 번 도시를 잇는 직통 열차의 소요 시간을 구해 보면 아래의 표와 같다.

$x \backslash y$	0	1	2	3	4
0	-	1	3	3	2
1	1	-	2	4	3
2	3	2	-	6	5
3	3	4	6	-	5
4	2	3	5	5	-

모든 가능한  $(x, y)$  순서쌍에 대해  $joy(x, y)$ 를 구해 보면 아래의 표와 같다.

$x \backslash y$	0	1	2	3	4
0	-	3	3	3	3
1	3	-	4	4	4
2	3	4	-	6	5
3	3	4	6	-	5
4	3	4	5	5	-

- $joy(0, 1) = 3$ 이다. 0번 도시에서 2번 도시로 직통 열차를 타고 이동하고, 2번 도시에서 4번 도시로 직통 열차를 타고 이동하고, 4번 도시에서 1번 도시로 직통 열차를 타고 이동하면 된다. 세 직통 열차의 소요 시간은 각각 3, 5, 4이다.
- $joy(1, 2) = 4$ 이다. 1번 도시에서 3번 도시로 직통 열차를 타고 이동하고, 3번 도시에서 2번 도시로 직통 열차를 타고 이동하면 된다. 두 통 열차의 소요 시간은 각각 4, 6이다.
- $joy(2, 3) = 6$ 이다. 2번 도시에서 3번 도시로 직통 열차를 타고 이동하면 된다. 이 직통 열차의 소요 시간은 6이다.

함수는 80을 반환해야 한다.

## 예제 2

$N = 5, U = [0, 0, 0, 0], V = [1, 2, 3, 4], W = [3, 2, 2, 1]$ 인 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
travel([0, 0, 0, 0], [1, 2, 3, 4], [3, 2, 2, 1])
```

모든 가능한  $(x, y)$  순서쌍에 대해  $\text{joy}(x, y)$ 를 구해 보면 아래의 표와 같다.

$x \backslash y$	0	1	2	3	4
0	-	3	3	3	3
1	3	-	5	5	4
2	3	5	-	5	4
3	3	5	5	-	4
4	3	4	4	4	-

함수는 78을 반환하여야 한다.

## 예제 3

$N = 6, U = [0, 1, 2, 3, 4], V = [1, 2, 3, 4, 5], W = [3, 1, 4, 1, 5]$ 인 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
travel([0, 1, 2, 3, 4], [1, 2, 3, 4, 5], [3, 1, 4, 1, 5])
```

모든 가능한  $(x, y)$  순서쌍에 대해  $\text{joy}(x, y)$ 를 구해 보면 아래의 표와 같다.

$x \backslash y$	0	1	2	3	4	5
0	-	11	10	8	9	14
1	11	-	10	8	9	11
2	10	10	-	8	9	10
3	8	8	8	-	8	8
4	9	9	9	8	-	9
5	14	11	10	8	9	-

함수는 284를 반환하여야 한다.

## Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1:  $N$
- Line  $2 + i$  ( $0 \leq i \leq N - 2$ ):  $U[i] V[i] W[i]$

Sample grader는 다음을 출력한다.

- Line 1: `travel`이 반환한 값

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.