



봉쇄 시간

헝가리에는 N 개의 도시가 있고, 각각 0부터 $N - 1$ 까지 번호가 매겨져 있다.

도시들은 $N - 1$ 개의 **양방향** 도로로 연결되어 있고, 도로들은 각각 0부터 $N - 2$ 까지 번호가 매겨져 있다. $0 \leq j \leq N - 2$ 인 모든 j 에 대해 도로 j 는 도시 $U[j]$ 와 도시 $V[j]$ 를 연결하고, 길이는 $W[j]$ 이다. 즉, 한 도시에서 다른 도시로 여행하는데 $W[j]$ 만큼의 시간이 걸린다. 각 도로는 두 다른 도시를 연결하며, 한 쌍의 도시는 최대 하나의 도로로 연결된다.

서로 다른 두 도시 a 와 b 를 잇는 **경로**는 다음 조건을 만족하는 서로 다른 도시들 p_0, p_1, \dots, p_t 이다.

- $p_0 = a$,
- $p_t = b$,
- 각각의 i ($0 \leq i < t$)에 대해서, 두 도시 p_i 와 p_{i+1} 를 연결하는 도로가 있다.

한 도시에서 다른 도시로 도로를 이용해서 여행하는 것은 항상 가능하다. 즉, 서로 다른 두 도시 사이에는 이를 연결하는 경로가 항상 있다. 또한 서로 다른 한 쌍의 도시마다 이 둘을 잇는 경로가 유일함을 보일 수 있다.

경로 p_0, p_1, \dots, p_t 의 **길이**는 경로상에 있는 연속한 도시들 사이를 잇는 t 개의 도로의 길이의 총합이다.

헝가리에서는 많은 사람들이 두 도시에서 벌어지는 국경일 행사를 참여하기 위해서 여행한다. 행사가 끝나면 여행자들은 집으로 돌아온다. 헝가리 정부는 여행자들이 주민들을 피곤하지 않게 하기 위한 방법을 고안하였다. 각 도시마다 헝가리 정부는 음이 아닌 정수인 **봉쇄 시간**을 정하려 한다. 또, 정부는 봉쇄 시간의 합은 K 이하여야 한다고 정했다. 보다 정확하게는, 0 이상 $N - 1$ 이하인 모든 i 에 대해서, 도시 i 에 할당된 봉쇄 시간을 음이 아닌 정수 $c[i]$ 라고 할 때, 모든 $c[i]$ 의 합은 K 이하여야 한다.

도시 a 가 있고, 각 도시들마다 봉쇄 시간이 정해졌다고 가정하자. 도시 b 가 도시 a 에서 **도달 가능**하다는 것은 $b = a$ 이거나, 두 도시 사이의 경로 p_0, \dots, p_t ($p_0 = a$ 이고 $p_t = b$)가 다음 조건들을 모두 만족한다는 것과 같은 뜻이다.

- 경로 p_0, p_1 의 길이가 최대 $c[p_1]$ 이며,
- 경로 p_0, p_1, p_2 의 길이가 최대 $c[p_2]$ 이며,
- ...
- 경로 $p_0, p_1, p_2, \dots, p_t$ 의 길이가 최대 $c[p_t]$ 이다.

올해 국경일 행사는 두 도시 X 와 Y 에서 치러진다. 각 도시마다 봉쇄 시간을 정했을 때, **편의성 점수**는 다음 두 수의 합으로 정의된다.

- 도시 X 에서 도달 가능한 도시의 수
- 도시 Y 에서 도달 가능한 도시의 수

어떤 도시가 도시 X 에서도 도달 가능하고, 도시 Y 에서도 도달 가능하다면 편의성 점수에 **두번** 더해진다는 사실에 유의하라.

당신이 할 일은 봉쇄 시간을 적절히 정해서 얻을 수 있는 편의성 점수의 최대값을 구하는 것이다.

Implementation Details

다음 함수를 구현해야 한다.

```
int max_score(int N, int X, int Y, int64 K, int[] U, int[] V, int[] W)
```

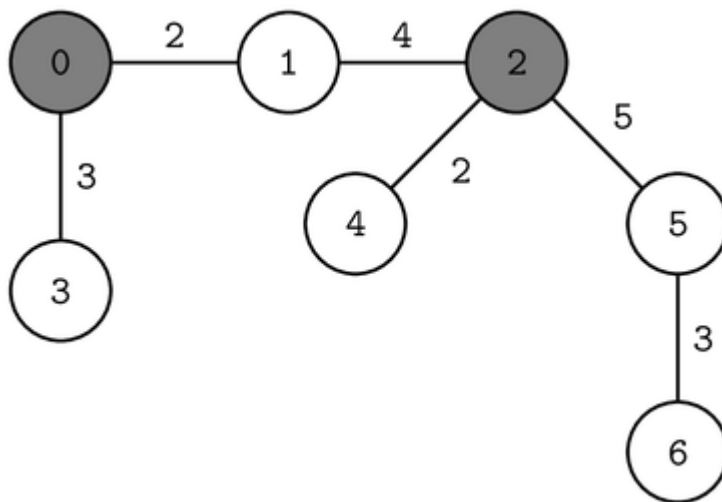
- N : 도시의 수
- X, Y : 국경일 행사가 치러지는 두 도시
- K : 봉쇄 시간의 합의 상한
- U, V : 도로 연결 정보를 나타내는 길이 $N - 1$ 인 배열
- W : 도로 길이를 나타내는 길이 $N - 1$ 인 배열
- 이 함수는 봉쇄 시간을 설정해서 얻을 수 있는 편의성 점수의 최대값을 리턴해야 한다.
- 이 함수는 하나의 테스트 케이스에서 **여러 번** 호출될 수 있다.

Example

다음 호출을 생각해보자.

```
max_score(7, 0, 2, 10,  
          [0, 0, 1, 2, 2, 5], [1, 3, 2, 4, 5, 6], [2, 3, 4, 2, 5, 3])
```

이는 다음 도로망에 대응한다.



봉쇄 시간이 다음과 같이 할당되었다고 가정하자.

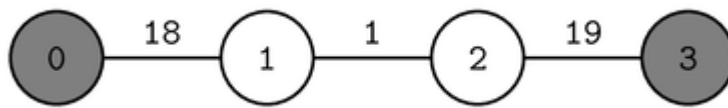
| 도시 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|---|---|---|---|---|---|---|
| 봉쇄 시간 | 0 | 4 | 0 | 3 | 2 | 0 | 0 |

봉쇄 시간의 총합은 9로, $K = 10$ 이하이다. 도시 0, 1, 3은 도시 X ($X = 0$)에서 도달 가능하며, 도시 1, 2, 4은 도시 Y ($Y = 2$)에서 도달 가능하다. 따라서 편의성 점수는 $3 + 3 = 6$. 어떻게 봉쇄 시간을 정하더라도, 편의성 점수를 6보다 크게 할 수 없다. 따라서 이 함수의 리턴값은 6이어야 한다.

또 다음 호출을 생각해보자.

```
max_score(4, 0, 3, 20, [0, 1, 2], [1, 2, 3], [18, 1, 19])
```

이는 다음 도로망에 대응한다.



봉쇄 시간이 다음과 같이 할당되었다고 가정하자.

| 도시 | 0 | 1 | 2 | 3 |
|-------|---|---|----|---|
| 봉쇄 시간 | 0 | 1 | 19 | 0 |

도시 0은 도시 X ($X = 0$)에서 도달 가능하며, 도시 2, 3은 도시 Y ($Y = 3$)에서 도달 가능하다. 따라서 편의성 점수는 $1 + 2 = 3$. 어떻게 봉쇄 시간을 정하더라도, 편의성 점수를 3보다 크게 할 수 없다. 따라서 이 함수의 리턴값은 3이어야 한다.

Constraints

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ ($0 \leq j \leq N - 2$ 인 각각의 j 에 대해서)
- $1 \leq W[j] \leq 10^6$ ($0 \leq j \leq N - 2$ 인 각각의 j 에 대해서)
- 도로를 이용해서 한 도시에서 다른 어떤 도시로도 항상 여행할 수 있다.
- $S_N \leq 200\,000$, 여기서 S_N 은 한 테스트 케이스에서 `max_score`의 모든 호출에 대한 N 값의 합이다.

Subtasks

$0 \leq j \leq N - 2$ 인 각각의 j 에 대해서 만약 도로 j 가 도시 j 와 도시 $j + 1$ 을 연결한다면 이 도로망이 **선형**이라고 한다.

1. (8 points) 도시 X 와 도시 Y 를 잇는 경로의 길이는 $2K$ 보다 크다.
2. (9 points) $S_N \leq 50$ 이고 도로망은 선형이다.
3. (12 points) $S_N \leq 500$ 이고 도로망은 선형이다.
4. (14 points) $S_N \leq 3\,000$ 이고 도로망은 선형이다.
5. (9 points) $S_N \leq 20$
6. (11 points) $S_N \leq 100$
7. (10 points) $S_N \leq 500$
8. (10 points) $S_N \leq 3\,000$
9. (17 points) 추가적인 제약조건이 없다.

Sample Grader

C 가 시나리오의 수, 즉 `max_score`를 호출한 횟수라고 하자. 샘플 그레이더는 다음 양식으로 입력을 읽는다.

- line 1: C

C 개의 시나리오에 대한 설명이 다음에 따라온다.

샘플 그레이더는 각 시나리오에 대한 정보를 다음 양식으로 읽는다.

- line 1: $N\ X\ Y\ K$
- line $2 + j$ ($0 \leq j \leq N - 2$): $U[j]\ V[j]\ W[j]$

각 시나리오에 대해서 샘플 그레이더는 한 줄을 다음 양식으로 출력한다.

- line 1: `max_score`의 리턴값