

## 외곽 순환 도로 2 (ringroad2)

KOI 도시는  $N$ 개의 교차로와  $N - 1$ 개의 양방향 도로로 이루어져 있으며, 임의의 서로 다른 두 교차로를 도로만을 사용하여 오갈 수 있다. 즉, 도시의 도로망은 트리 구조를 이룬다. 도로는 2차원 평면 상에 있으며, 두 도로가 끝점을 제외한 위치에서 교차하지 않는다. 각 도로에는 0 이상의 정수 가중치가 존재한다.

KOI 도시는 몇십 년 전까지만 하여도 작은 마을이었지만, 사람이 유입되고 크기가 커지면서 급격히 팽창하기 시작하였다. 급격한 팽창 과정에서 시장은 행정 편의를 위해 교차로에 0 이상  $N - 1$  이하의 번호를 매겼다. 이 때 번호 시스템은 다음과 같은 성질을 만족한다.

- 0번 교차로는 도시의 중심으로, 2개 이상의 도로와 인접함이 보장된다.
- 교차로에 매겨진 번호는 0번 교차로를 루트로 한 트리의 전위 순회 순서 (Preorder) 중 하나이다.
- 모든 교차로에 대해서, 인접한 (도로로 직접 연결된) 교차로 중 가장 번호가 작은 교차로를 생각해 보자. 이 교차로를 기준으로 반시계 방향으로 인접한 교차로의 번호를 나열했을 때, 번호는 증가한다.

KOI 도시에 많은 사람들이 유입되면서, 교통 체증 문제가 심화되었다. 시장은 이를 해결하기 위해 교통 인프라가 가장 빈약한 도시들을 외곽 순환 도로로 이었다. KOI 도시에 있는 모든 교차로 중, 단 1개의 도로와 인접한 교차로들의 번호를 증가하는 순서대로 나열한 리스트를  $\{V[0], V[1], \dots, V[K - 1]\}$ 라고 하자. 시장은, 모든  $0 \leq i < K - 1$ 에 대해서,  $V[i]$ 번 교차로와  $V[(i + 1) \bmod K]$ 번 교차로를 잇는 양방향 도로를 건설하였다. 각 도로의 가중치는 0 이상의 정수  $W[i]$ 이다. 번호 시스템의 특성상, 두 도로가 끝점을 제외한 위치에서 교차하지 않게끔 외곽 순환 도로를 이을 수 있음을 관찰하면 좋다.

KOI 도시를 거점으로 하는 사악한 범죄단 Dlalswp은 시민들에게 많은 피해를 끼치고 있다. 시장은 Dlalswp 범죄단을 잡기 위해 정보요원들을 투입하였다. 정보요원들의 첩보에 의하면, Dlalswp 범죄단은 길이가 홀수인 단순 사이클을 범죄 영역으로 설정하고, 해당 사이클을 돌면서 범죄 행위를 저지른다.

악명높은 Dlalswp 범죄단을 검거하기 위해, 시장은 모든 길이가 홀수인 단순 사이클에 경찰이 투입된 간선이 하나 이상 있도록 KOI 도시의 도로에 경찰을 투입하려고 한다. 각 도로에 경찰을 투입하기 위해 필요한 비용은 도로의 가중치와 동일하다. 시장이 목표를 이루기 위해 필요한 최소 비용을 계산하여라.

## 함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
long long place_police(vector<int> P, vector<long long> C, vector<long long> W)
```

- 이 함수는 단 한 번만 호출된다.
- $P$ : 크기가  $N - 1$ 인 정수 배열. 외곽 순환 도로를 짓기 전 원래 KOI 도시를 이루는 도로들을 나타낸다. 모든  $i$  ( $0 \leq i \leq N - 2$ )에 대해,  $P[i]$ 번 교차로와  $i + 1$ 번 교차로를 잇는 도로가 있다.
- $C$ : 크기가  $N - 1$ 인 정수 배열. 모든  $i$  ( $0 \leq i \leq N - 2$ )에 대해,  $P[i]$ 번 교차로와  $i + 1$ 번 교차로를 잇는 도로의 가중치는  $C[i]$ 이다.
- $W$ : 크기가  $K$ 인 정수 배열. 모든  $i$  ( $0 \leq i \leq K - 1$ )에 대해,  $V[i]$ 번 교차로와  $V[(i + 1) \bmod K]$ 번 교차로를 잇는 양방향 도로의 가중치는  $W[i]$ 이다.

- 이 함수는 모든 길이가 홀수인 단순 사이클에 경찰이 투입된 간선이 하나 이상 있도록 KOI 도시의 도로에 경찰을 투입하는 최소 비용을 반환해야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

## 제약 조건

- $4 \leq N \leq 100\,000$
- 모든  $i$ 에 대해  $0 \leq P[i] \leq i$  ( $0 \leq i \leq N - 2$ )
- 모든  $i$ 에 대해  $0 \leq C[i] \leq 10^{12}$  ( $0 \leq i \leq N - 2$ )
- 모든  $i$ 에 대해  $0 \leq W[i] \leq 10^{12}$  ( $0 \leq i \leq K - 1$ )

## 부분문제

1. (6점)
  - $K \leq 5$
2. (8점)
  - 모든  $i$ 에 대해  $P[i] = 0$  ( $0 \leq i \leq N - 2$ )
3. (5점)
  - 모든  $i$ 에 대해  $C[i] \leq 10^6$  ( $0 \leq i \leq N - 2$ )
  - 모든  $i$ 에 대해  $W[i] = 10^{12}$  ( $0 \leq i \leq K - 1$ )
  - $K$ 는 짝수
4. (15점)
  - 모든  $i$ 에 대해  $C[i] \leq 10^6$  ( $0 \leq i \leq N - 2$ )
  - 모든  $i$ 에 대해  $W[i] = 10^{12}$  ( $0 \leq i \leq K - 1$ )
5. (57점)
  - 4개 이상의 도로와 인접한 교차로가 존재하지 않음.
6. (9점)
  - 추가적인 제약 조건이 없다.

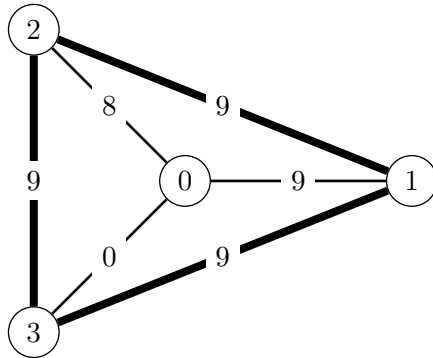
## 예제 1

$N = 4$ ,  $P = [0, 0, 0]$ ,  $C = [9, 8, 0]$ ,  $W = [9, 9, 9]$ 인 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
place_police([0, 0, 0], [9, 8, 0], [9, 9, 9])
```

아래 그림은 KOI 도시의 지도를 나타낸다.



KOI 도시의 홀수 사이클은 총 4개로,  $\{0, 1, 2\}$ ,  $\{0, 2, 3\}$ ,  $\{0, 3, 1\}$ ,  $\{1, 2, 3\}$ 이 있다. 시장이  $(0, 3)$ 을 잇는 가중치 0의 도로와,  $(1, 2)$ 를 잇는 가중치 9의 도로에 경찰을 배치할 경우, 모든 홀수 사이클에 경찰이 배치된 도로가 하나 이상 존재한다. 이 배치의 비용은  $0 + 9 = 9$ 이며, 이보다 더 적은 비용으로 목표를 달성할 수 없다.

함수는 9를 반환해야 한다.

## 예제 2

$N = 11$ ,

$P = [0, 0, 2, 3, 3, 2, 0, 7, 7, 9]$ ,

$C = [9, 8, 0, 7, 1, 6, 0, 7, 1, 6]$ ,

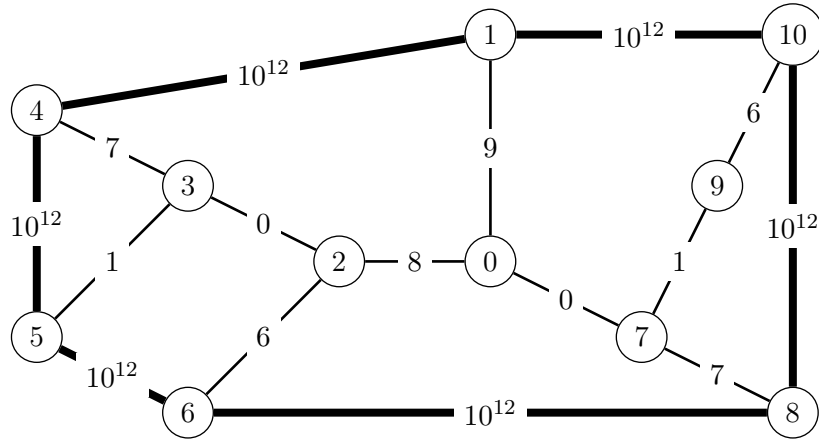
$W = [1000000000000, \dots, 1000000000000]$

인 경우를 생각해 보자.  $W$ 의 크기는 6이고 모든 원소는  $10^{12}$ 이다.

그레이더는 다음 함수를 호출한다.

```
place_police([0, 0, 2, 3, 3, 2, 0, 7, 7, 9],
             [9, 8, 0, 7, 1, 6, 0, 7, 1, 6],
             [1000000000000, ..., 1000000000000])
```

아래 그림은 KOI 도시의 지도를 나타낸다.



시장이 (2, 3)을 잇는 가중치 0의 도로와, (0, 7)을 잇는 가중치 0의 도로와, (3, 5)를 잇는 가중치 1의 도로에 경찰을 배치할 경우, 모든 홀수 사이클에 경찰이 배치된 도로가 하나 이상 존재한다. 이 배치의 비용은  $0 + 0 + 1 = 1$ 이며, 이보다 더 적은 비용으로 목표를 달성할 수 없다.

함수는 1을 반환해야 한다.

### Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1:  $N$
- Line  $2 + i$  ( $0 \leq i \leq N - 2$ ):  $P[i] C[i]$
- Line  $1 + N$ :  $W[0] W[1] \cdots W[K - 1]$

Sample grader는 다음을 출력한다.

- Line 1: 함수 `place_police`가 반환한 값

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.