

택시 여행 (taxi)

IOI 나라는 N 개의 도시와 도시들을 잇는 $N - 1$ 개의 양방향 도로로 이루어져 있으며, 임의의 서로 다른 두 도시를 도로만을 사용하여 오갈 수 있다. 즉, IOI 나라의 도로망은 트리 구조를 이룬다.

도시에는 각각 0 이상 $N - 1$ 이하의 서로 다른 번호가 붙어 있으며, 0번 도시가 IOI 나라의 수도이다. 또 모든 $0 \leq i \leq N - 2$ 에 대해서 i 번 도로는 $U[i]$ 번 도시와 $V[i]$ 번 도시를 연결하며 도로의 길이는 $W[i]$ km 이다.

IOI 나라에서는 도시별로 택시의 운임이 다르다. 구체적으로, 모든 $0 \leq i \leq N - 1$ 에 대해 i 번 도시에서 출발하는 택시는 기본 요금 $A[i]$ 원과 거리 당 요금 $B[i]$ 원을 가진다. 이는 i 번 도시에서 택시를 타고 출발하여 d km 만큼 이동할 경우 $A[i] + d \times B[i]$ 원을 내야 함을 뜻한다.

서현이는 현재 수도인 0번 도시에 살고 있다. 서현이는 다른 도시들로 택시를 타고 여행을 떠나려고 한다. 서현이가 어떤 도시에 도착했을 때, 서현이는 타던 택시를 계속 타거나 그 도시에서 출발하는 택시로 갈아탈 수 있다. 물론 택시를 갈아타면 기본 요금을 내야 하며 거리 당 요금도 변할 수 있다. 0번 도시에서 출발하여 다른 모든 도시들로 가는 데 필요한 최소 비용을 각각 계산하여라.

함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

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

- 이 함수는 단 한 번만 호출된다.
- A : 크기가 N 인 정수 배열. 모든 i ($0 \leq i \leq N - 1$)에 대해, $A[i]$ 는 i 번 도시에서 출발하는 택시의 기본 요금이다.
- B : 크기가 N 인 정수 배열. 모든 i ($0 \leq i \leq N - 1$)에 대해, $B[i]$ 는 i 번 도시에서 출발하는 택시의 이동 거리(km)당 요금이다.
- 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 $W[i]$ km의 도로가 있다.
- 이 함수는 크기가 $N - 1$ 인 배열 C 를 반환해야 한다. 모든 i ($0 \leq i \leq N - 2$)에 대해, $C[i]$ 는 0번 도시에서 출발해 $i + 1$ 번 도시까지 가는 데 필요한 최소 비용과 같아야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

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

부분문제

- (7점)
 - $N \leq 20$
- (8점)
 - 모든 i 에 대해 $U[i] = i; V[i] = i + 1$ ($0 \leq i \leq N - 2$)
- (13점)
 - $N \leq 2\,000$
- (17점)
 - 모든 i 에 대해 $B[i] \leq 30$ ($0 \leq i \leq N - 1$)
- (29점)
 - $B[i] \neq 0$ 인 i 가 2000개 이하. ($0 \leq i \leq N - 1$)
- (26점)
 - 추가적인 제약 조건이 없다.

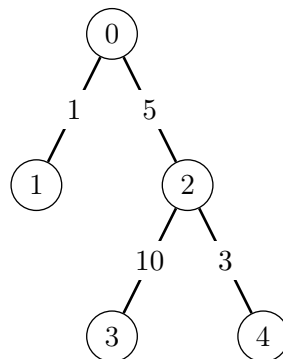
예제 1

$N = 5, A = [10, 5, 13, 4, 3], B = [10, 7, 5, 9, 1], U = [1, 0, 3, 2], V = [0, 2, 2, 4], W = [1, 5, 10, 3]$ 인 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
travel([10, 5, 13, 4, 3], [10, 7, 5, 9, 1], [1, 0, 3, 2], [0, 2, 2, 4], [1, 5, 10, 3])
```

아래 그림은 IOI 나라의 지도를 나타낸다.



- 0번 도시에서 1번 도시로 이동할 때는 0번 도시에서 택시를 타서 1번 도시로 가는 방법이 최적이며 이때 총 비용은 20원이다.
- 0번 도시에서 2번 도시로 이동할 때는 0번 도시에서 택시를 타서 2번 도시로 가는 방법이 최적이며 이때 총 비용은 60원이다.
- 0번 도시에서 4번 도시로 이동할 때는 0번 도시에서 택시를 타서 1번 도시로 간 다음, 택시를 갈아타고 다시 0번과 2번 도시를 거쳐 4번 도시로 가는 방법이 최적이며 이때 총 비용은 88원이다.
- 0번 도시에서 3번 도시로 이동할 때는 0번 도시에서 택시를 타서 1번 도시로 간 다음, 택시를 갈아타고 0번과 2번 도시를 거쳐 4번 도시로 간 다음, 택시를 다시 갈아타고 2번 도시를 거쳐 3번 도시로 가는 방법이 최적이며 이때 총 비용은 104원이다.

함수는 [20, 60, 104, 88]을 반환해야 한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: N
- Line 2: $A[0] A[1] \dots A[N - 1]$
- Line 3: $B[0] B[1] \dots B[N - 1]$
- Line $4 + i$ ($0 \leq i \leq N - 2$): $U[i] V[i] W[i]$

Sample grader는 다음을 출력한다.

- Line i : `travel`이 반환한 배열의 i 번째 원소

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.