바이러스 (virus)

KOI 도시는 지난 코로나19 대유행으로 많은 피해를 겪었기 때문에, 향후 발생할 수 있는 팬데믹 상황에 철저히 대비하려고 한다. 이를 위해, KOI 도시는 현재의 도시 구조가 바이러스에 어느 정도로 취약한지를 분석하고자 한다.

KOI 도시는 N 개의 지점과 $N-1$ 개의 양방향 도로로 이루어져 있으며, 임의의 서로 다른 두 지점을 도로만을 사용하여 오갈 수 있다. 즉, 도시의 도로망은 트리 구조를 이룬다. 각 지점은 0 이상 $N-1$ 이하의 서로 다른 정수로 구분된다. 도시의 도로망이 트리이기 때문에, 두 지점 u, v 에 대해서, u번 지점에서 v번 지점으로 이동하는 단순 경로는 유일하다. 이 유일한 경로의 간선의 개수를 u 와 v 의 **거리** 라고 정의하자.

KOI 도시에는 M 명의 사람들이 살고 있다. 모든 $0 \le j \le M-1$ 에 대해, j 번 사람은 $P[j]$ 번 지점에 살고 있으며, 해당 지점에서 거리가 $D[j]$ 이하인 지점을 오갈 수 있다.

KOI 도시의 바이러스학자들은, 두 사람 간에 바이러스가 전파되는 과정을 다음과 같이 모델링하였다. 모든 $0 \le v \le N-1$ 에 대해, v번 지점의 **전파 시간**은 $C[v]$ 라는 양의 정수로 표현된다. j 번 사람이 시각 t 에 처음으로 바이러스에 감염되었다고 하고, j 번 사람에게서 바이러스를 전파받을 사람을 k 번 사람이라고 하자. w번 지점을 j번 사람과 k번 사람이 공동으로 오갈 수 있다면 – 다시 말해, w번 지점과 $P[j]$번 지점의 거리가 $D[j]$ 이하고 w번 지점과 $P[k]$번 지점의 거리가 $D[k]$ 이하라면, w번 지점은 **전파의 매개체**가 된다.

만약에 전파의 매개체가 되는 지점이 없다면, k번 사람은 j번 사람으로부터 직접 바이러스에 감염되지 않는다. (물론 다른 사람을 통하여 간접적으로 감염될 수는 있다) 전파의 매개체가 되는 지점이 있다면, 그러한 지점 중 전파 시간을 **최소화** 하는 지점의 번호를 x 라고 하자. k 번 사람이 만약 시각 $t + C[x]$ 에 바이러스에 감염되지 않았다면, k 번 사람은 그 시각에 j 번 사람에 의해 바이러스에 감염된다. 바이러스는, 모든 서로 다른 사람의 쌍 $0 \le j, k \le M-1, j \ne k$ 에 대해 이러한 식으로 확산한다.

위와 같은 모델링 하에서, KOI 도시의 연구진들은 0 번 사람이 시각 0 에 바이러스에 감염되었을 때, 다른 사람들이 바이러스에 언제 감염되는지를 계산하려고 한다. 당신은, 모든 $0 \le j \le M-1$ 에 대해, j 번 사람이 처음으로 바이러스에 감염되는 시각을 계산해야 한다. 단, 만약 j 번 사람이 바이러스에 감염되지 않는다면, 그 시각을 -1 이라고 기록해야 한다.

함수 목록 및 정의

여러분은 아래 함수를 구현해야 한다.

```
vector<long long> find_spread(int N, int M, vector<int> A, vector<int> B,
vector<int> P, vector<int> D, vector<int> C)
```

- N: KOI 도시의 지점의 수.
- M: KOI 도시의 사람의 수.
- A, B: 길이가 $N-1$ 인 배열. 모든 i $(0 \le i \le N-2)$에 대해, $A[i]$ 번 지점과 $B[i]$ 번 지점을 잇는 도로가 존재한다. 두 배열에서 모든 도로는 정확히 한 번씩 등장한다.
- P, D: 길이가 M 인 배열. 모든 j $(0 \le j \le M-1)$에 대해, j 번 사람은 $P[j]$ 번 지점에 살고 있으며, 해당 지점에서 거리가 $D[j]$ 이하인 지점을 오갈 수 있다.

- C: 길이가 N 인 배열. 모든 v $(0 \le v \le N-1)$에 대해, v 번 지점의 전파 시간은 $C[v]$ 이다.
- 이 함수는 크기가 M 인 배열 T를 반환해야 한다. 모든 j $(0 \le j \le M-1)$에 대해, $T[j]$는 j 번 사람이 바이러스에 감염된다면 처음 바이러스에 감염되는 시각이어야 하고, 감염되지 않는다면 -1 이어야 한다.
- 이 함수는 매 테스트 케이스마다 단 한 번 호출된다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $1 \le N \le 100\,000$
- $1 \le M \le 100\,000$
- 모든 $0 \le i \le N-2$에 대해 $0 \le A[i], B[i] \le N-1$, $A[i] \ne B[i]$
- 모든 $0 \le j \le M-1$에 대해 $0 \le P[j], D[j] \le N-1$
- 모든 $0 \le v \le N-1$에 대해 $1 \le C[v] \le 10^9$

부분문제

1. (5점)
    - $N \le 500$, $M \le 500$
    - 모든 $0 \le i \le N-2$에 대해 $A[i] = i, B[i] = i+1$.
2. (8점)
    - $N \le 5\,000$, $M \le 5\,000$
    - 모든 $0 \le i \le N-2$에 대해 $A[i] = i, B[i] = i+1$.
3. (27점)
    - 모든 $0 \le i \le N-2$에 대해 $A[i] = i, B[i] = i+1$.
4. (5점)
    - $N \le 500$, $M \le 500$
5. (8점)
    - $N \le 5\,000$, $M \le 5\,000$
6. (47점)
    - 추가적인 제약 조건이 없다.

예제 1

$N = 8$, $M = 5$, $A = [0, 1, 2, 3, 4, 5, 6]$, $B = [1, 2, 3, 4, 5, 6, 7]$, $P = [2, 5, 7, 1, 4]$, $D = [2, 0, 0, 1, 1]$, $C = [40, 5, 5, 16, 32, 8, 1, 10]$ 인 경우를 생각해 보자.

그레이더는 다음과 같이 함수를 호출한다.

```
find_spread(8, 5, [0, 1, 2, 3, 4, 5, 6], [1, 2, 3, 4, 5, 6, 7], [2, 5, 7, 1, 4], [2, 0, 0, 1, 1], [40, 5, 5, 16, 32, 8, 1, 10])
```

이 경우 바이러스는 다음과 같은 방법으로 전파한다.

- 시간 0에 0번 사람이 감염된다.
- 시간 5에 3번 사람이 감염된다. 3번 사람은 0번 사람에게서 전염되었으며, 이 때 전파의 매개체는 1번 지점이다. (2번 지점 역시 전파의 매개체가 될 수 있다.)
- 시간 16에 4번 사람이 감염된다. 4번 사람은 0번 사람에게서 전염되었으며, 이 때 전파의 매개체는 3번 지점이다.
- 시간 24에 1번 사람이 감염된다. 1번 사람은 4번 사람에게서 전염되었으며, 이 때 전파의 매개체는 5번 지점이다.
- 2번 사람은 끝까지 감염되지 않는다.

함수는 $[0, 24, -1, 5, 16]$ 을 반환해야 한다.

예제 2

그레이더는 다음과 같이 함수를 호출한다.

```
find_spread(8, 5, [0, 1, 2, 3, 4, 3, 3], [1, 2, 3, 4, 5, 6, 7], [2, 5, 7, 1, 4], [2, 0, 0, 1, 1], [40, 5, 5, 16, 32, 8, 1, 10])
```

함수는 $[0, 24, 10, 5, 16]$ 을 반환해야 한다.

예제 3

그레이더는 다음과 같이 함수를 호출한다.

```
find_spread(10, 10, [9, 0, 1, 5, 3, 8, 0, 6, 0], [3, 6, 8, 1, 2, 7, 4, 5, 9], [9, 7, 1, 8, 6, 1, 4, 7, 5, 5], [0, 2, 0, 1, 3, 2, 6, 2, 1, 4], [10, 12, 5, 9, 8, 21, 20, 6, 13, 5])
```

함수는 $[0, 11, 17, 11, 5, 11, 5, 11, 17, 5]$ 을 반환해야 한다.

예제 4

그레이더는 다음과 같이 함수를 호출한다.

```
find_spread(1, 2, [], [], [0, 0], [0, 0], [1000000000])
```

함수는 $[0, 1000000000]$ 을 반환해야 한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: N M
- Line $2 + i$ $(0 \leq i \leq N - 2)$: $A[i]$ $B[i]$
- Line $1 + N + j$ $(0 \leq j \leq M - 1)$: $P[j]$ $D[j]$
- Line $1 + N + M$: $C[0]$ $C[1]$ $\ldots$ $C[N - 1]$

Sample grader는 다음을 출력한다.

- Line $1 + j$ $(0 \leq j \leq M - 1)$: 함수 `find_spread`가 반환한 배열의 j번 원소

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.