



2019 숭고한 연합

# Algorithm Contest

- 중급반 -

대회가 시작되기 전까지 절대 표지를 넘기지 마세요

< Hosted by >



< Sponsored by >



SAMSUNG  
SOFTWARE  
MEMBERSHIP

고려대학교 SW 중심대학사업단



대회는 총 7문제이며, 구성은 다음과 같습니다. 총 문제지가 표지를 제외하고 14쪽인지 확인하시길 바랍니다.

- A 이건 꼭 풀어야 해!
- B 무한부스터
- C 이진수 변환
- D 백도어
- E FLEX
- F 통신망 분할
- G 트리의 외심

- 대회 시간은 4시간입니다.
- 대회가 진행되는 동안 대회 사이트를 제외한 웹에 접속/열람/검색 등을 할 수 없습니다. 예외로, 공지를 통해 명시된 레퍼런스 사이트는 열람할 수 있습니다.
- 이전에 저장해두었던 자료/코드 등을 열람 및 사용할 수 없습니다. 예외로, 주최진 측에서 제공하는 ‘템플릿 코드 문서’는 열람할 수 있습니다.
- 그 외에 운영진이 판단하기에 악의적인 제출을 진행하거나 부정 행위를 저지를 경우 경고 없이 대회 참가 자격이 박탈될 수 있습니다.
- 컴파일 환경은 BOJ의 기본 환경을 따릅니다.
- 대회 중 ‘질문’ 탭을 통해 주최진에게 물을 수 있습니다. 예/아니오로 답이 나오는 질문을 해주시기 바랍니다.
- 문제지는 참가자의 편의를 위해 제공되었을 뿐, 웹사이트의 내용이 공식 문제입니다. 인쇄본과 사이트의 내용 중 상이한 부분이 있을 경우, 사이트에 기재된 설명을 참고하시길 바랍니다.

## 문제 A. 이걸 꼭 풀어야 해!

시간 제한: 1 초

메모리 제한: 512 MB

송실꿀 높은 언덕 깊은 골짜기에 출제로 고통 받는 옥제가 살고 있다!

옥제는 또 출제를 해야 해서 단단히 화가 났다. 그래서 옥제는 길이  $N$ 짜리 수열  $A$ 를 만들고,  $A$ 를 비내림차순으로 정렬해서 수열  $B$ 를 만들어 버렸다!! 여기서  $B$ 를 출력하기만 하면 문제가 너무 쉬우니까 하나만 더 하자. 아래와 같은 질문이 무려  $Q$ 개나 주어진다!! (ㅎㅎ;; 스스.. ㅋㅋ!!)

- **L R**:  $B_L + B_{L+1} + \dots + B_{R-1} + B_R$  을 출력한다.



Figure 1: 모든 참가자가 문제를 풀 수 있을 것이라고 기대하는 옥제의 표정

옥제의 질문에 답하고 함께 엠티를 떠나자!!

### 입력 형식

첫 번째 줄에 수열  $A$ 의 길이  $N$ 과 질문의 개수  $Q$ 가 공백으로 구분되어 주어진다. ( $1 \leq N, Q \leq 300,000$ )

두 번째 줄에  $N$ 개의 정수  $A_1, A_2, \dots, A_N$  이 공백으로 구분되어 주어진다.  $A_i$ 는 수열  $A$ 의  $i$ 번째 수이다. ( $1 \leq A_i \leq 1,000$ )

세 번째 줄부터  $Q$ 개의 줄에 걸쳐 옥제의 질문을 의미하는 두 수  $L, R$ 이 공백으로 구분되어 주어진다. ( $1 \leq L \leq R \leq N$ )

주어지는 모든 입력은 자연수이다.

### 출력 형식

$Q$ 개의 줄에 걸쳐, 질문의 답을 순서대로 각각 출력한다.

## 예제

| 표준 입력  | 표준 출력                        |
|--|------------------------------|
| 5 6<br>2 5 1 4 3<br>1 5<br>2 4<br>3 3<br>1 3<br>2 5<br>4 5 | 15<br>9<br>3<br>6<br>14<br>9 |
| 5 3<br>2 5 1 2 3<br>1 3<br>2 3<br>1 5                      | 5<br>4<br>13                 |

## 설명

첫 번째 예제에서 [2, 5, 1, 4, 3]을 비내림차순으로 정렬하면 [1, 2, 3, 4, 5]이다.

두 번째 예제에서 [2, 5, 1, 2, 3]을 비내림차순으로 정렬하면 [1, 2, 2, 3, 5]이다.

## 참고 사항

비내림차순은 원소가 감소하지 않는 (같거나 증가하는) 순서를 말한다.

```
while (Q--) {
    int sum = 0, L, R;
    scanf("%d %d", &L, &R);
    for (int i = L; i <= R; i++) {
        sum += a[i];
    }
    printf("%d\n", sum);
}
```

위와 같이 각 질문마다 반복문을 매번 돌려서 답을 계산하면, 시간복잡도가  $O(QN)$  이 되므로 시간 초과를 받게 된다. 다른 방법을 이용해 문제를 해결해야 한다.

## 문제 B. 무한부스터

시간 제한: 1 초

메모리 제한: 512 MB

카트라이더를 처음 시작하는 카린이 정범이는 어려운 조작법에 실망감이 커져가고 있다. 드리프트, 순간 부스터, 커팅, 톱톡이 등등 어려운 테크닉에 질린 정범이는 그나마 쉬운 ‘송고한 무한부스터 모드’에 도전해보려고 한다.

‘송고한 무한부스터 모드’는 크기  $N \times M$  의 직사각형 모양의 맵에서 진행되며, 맵 전체가 단위 격자로 구성되어 있다. 기존의 ‘무한부스터 모드’와는 다르게, 모든 격자 안에는 특정 개수의 부스터 아이템이 위치한다. 이 모드에서 플레이의 방식은 다음과 같다.

처음에 플레이어의 카트바디는 출발지점인 1행 1열에 위치하며, 멈춰 있는 상태이고, 보유하고 있는 부스터 아이템의 개수는 0개이다. 목표는 도착지점인  $N$ 행  $M$ 열의 격자에 도달하는 것이며, 도달하는 즉시 게임이 종료된다. 카트바디가 격자에 멈추어 있을 때, 격자에 놓여있는 부스터 아이템을 자동으로 전부 습득하게 된다. 이 과정에서  $x$ 개를 습득했다면 한 방향을 정해 오른쪽으로 최대  $x$ 칸을 가거나, 아래쪽으로 최대  $x$ 칸을 이동할 수 있으며, 1칸 단위로 이동하게 된다. 예를 들어 부스터 아이템을 3개 습득했을 때, 오른쪽으로 2칸 이동이나 아래쪽으로 3칸 이동은 가능하지만, 오른쪽으로 1칸 이동 후 아래로 2칸 이동이나 왼쪽으로 1칸 이동이나 아래쪽으로 2.718칸 이동은 불가능하다. **이동 후 멈추면서 보유하고 있던 부스터 아이템은 모두 소진된다.**

이동중에 멈추지 않고 지나치는 격자의 부스터 아이템은 습득할 수 없으며, 카트바디는 맵을 벗어나는 방향으로 움직일 수 없다.

정범이는 ‘송고한 무한부스터 모드’에서 출발지점부터 도착지점까지 주행하면서 부스터 아이템을 획득하게 되는 격자의 개수를 최소화하고 싶다. 카린이 정범이를 도와주도록 하자.

### 입력 형식

첫 번째 줄에 맵의 세로 길이와 가로 길이를 나타내는 양의 정수  $N$ 과  $M$ 이 공백으로 구분되어 주어진다. ( $1 \leq N, M \leq 300$ )

두 번째 줄부터  $N$ 개의 줄에 걸쳐 각 격자에 있는 부스터 아이템 개수인  $M$ 개의 양의 정수  $a_{ij}$ 가 공백으로 구분되어 주어진다. ( $1 \leq a_{ij} \leq \max(N, M)$ )  $a_{ij}$ 는  $i$ 행  $j$ 열의 격자에 있는 부스터 아이템 개수이다.

출발지점과 도착지점은 다르다.

### 출력 형식

첫 번째 줄에 정범이가 맵의 출발지점부터 도착지점까지 이동하면서 부스터 아이템을 획득하게 되는 격자의 최소 개수를 출력한다.

## 예제

| 표준 입력  | 표준 출력 |
|--|-------|
| 4 5<br>1 1 4 1 3<br>3 4 1 3 2<br>1 1 5 3 2<br>5 3 1 1 1    | 3     |
| 5 4<br>2 4 2 3<br>1 1 1 3<br>2 1 2 2<br>1 4 4 1<br>1 2 2 1 | 3     |

## 설명

첫 번째 예제에서 3번 만에 이동하는 최적 경로는 다음과 같다.

- 1행 1열에서 1개의 부스터 아이템을 획득하고, 아래쪽으로 1칸 이동하여 2행 1열에서 멈춘다.
- 2행 1열에서 3개의 부스터 아이템을 획득하고, 아래쪽으로 2칸 이동하여 4행 1열에서 멈춘다.
- 4행 1열에서 5개의 부스터 아이템을 획득하고, 오른쪽으로 4칸 이동하여 4행 5열에서 멈춘다. 도착지점에 도달했기 때문에 게임이 종료된다.

두 번째 예제에서 3번 만에 이동하는 최적 경로는 다음과 같다.

- 1행 1열에서 2개의 부스터 아이템을 획득하고, 오른쪽으로 1칸 이동하여 1행 2열에서 멈춘다.
- 1행 2열에서 4개의 부스터 아이템을 획득하고, 아래쪽으로 4칸 이동하여 5행 2열에서 멈춘다.
- 5행 2열에서 2개의 부스터 아이템을 획득하고, 오른쪽으로 2칸 이동하여 5행 4열에서 멈춘다. 도착지점에 도달했기 때문에 게임이 종료된다.

## 문제 C. 이진수 변환

시간 제한: 1 초

메모리 제한: 512 MB

당신에게 자연수  $x_0$ 와  $N$ 이 주어졌다. 지금부터 당신은 이 자연수  $x_0$ 를  $N$ 번의 ‘변환’을 통해 0으로 바꿀 것이다. 변환이란, 양의 정수를 이진법으로 표기했을 때, 1개 이상의 1을 0으로 바꾸는 작업이다. 예를 들어 9를 이진법으로 나타내면  $1001_{(2)}$  인데, 9는  $0(0000_{(2)})$ ,  $1(0001_{(2)})$ , 또는  $8(1000_{(2)})$ 로 변환될 수 있다 (바뀐 자릿수는 밑줄로 표기되었다). 여러분의 목표는  $x_i$ 를 변환하여  $x_{i+1}$ 를 만드는 과정을 반복해,  $x_N$ 을 0으로 만드는 것이다.

위 조건을 만족하는 수열  $X = [x_0, x_1, x_2, \dots, x_N]$ 는 존재하지 않을 수도 있지만, 여러 개가 존재할 수도 있다. 만약 존재한다면, 각 수열별로 인접한 원소들의 차들의 집합  $D(X) = \{x_0 - x_1, x_1 - x_2, \dots, x_{N-1} - x_N\}$ 를 정의하자. 이 집합의 원소들의 최대값과 최소값의 차이를 최소화하도록, 수열  $X$ 를 만들려고 한다. 즉, 가능한 모든 수열  $X_i$  중 ( $D(X_i)$ 에 속한 원소의 최댓값 -  $D(X_i)$ 에 속한 원소의 최솟값)이 최소가 되는  $X_i$ 를 찾고자 한다.

이상해보일 수 있는 문제지만, 당신은 대답해야 한다. 과연 1초 안에 답할 수 있을까?

### 입력 형식

첫 번째 줄에 변환할 자연수와 변환 횟수를 의미하는 두 자연수  $x_0$ 과  $N$ 이 공백으로 구분되어 주어진다. ( $1 \leq x_0 \leq 10^{16}, 1 \leq N \leq 50$ )

### 출력 형식

만약 조건을 만족하는 수열이 존재하지 않으면 첫 번째 줄에 -1을 출력한다.

조건을 만족하는 수열이 존재한다면, 수열의 원소를 의미하는  $N$ 개의 정수  $x_1, x_2, \dots, x_N$ 을 공백으로 구분하여 출력한다.

조건을 만족하는 수열이 여러 개 존재한다면, 아무 것이나 출력해도 좋다.

### 예제

| 표준 입력 | 표준 출력 |
|-------|-------|
| 23 2  | 16 0  |
| 48 5  | -1    |

### 설명

첫 번째 예제에서  $x_0 = 23 = 10111_{(2)}$ ,  $x_1 = 16 = 10000_{(2)}$ ,  $x_2 = 0 = 00000_{(2)}$ 로 수열을 구성하면 인접한 원소의 차의 최대값과 최소값의 차이는  $16 - 7 = 9$ 가 되며, 이보다 작게 만들 수 있는 방법은 존재하지 않는다. 또 다른 답으로는  $x_0 = 23$ ,  $x_1 = 7$ ,  $x_2 = 0$ 이 있다.

두 번째 예제에서 주어진 조건을 만족하는  $x_1, x_2, x_3, x_4, x_5$ 는 존재하지 않는다.

## 문제 D. 백도어

시간 제한: 2 초

메모리 제한: 512 MB

유섭이는 무척이나 게으르다. 오늘도 할 일을 모두 미뤄둔 채 열심히 물을 하던 유섭이는 오늘까지 문제를 내야 한다는 사실을 깨달았다. 그러나 게임은 시작되었고 지는 걸 무척이나 싫어하는 유섭이는 어쩔 수 없이 백도어를 해 게임을 최대한 빠르게 끝내기로 결심하였다.

최대한 빨리 게임을 끝내고 문제를 출제해야 하기 때문에 유섭이는 최대한 빨리 넥서스가 있는 곳으로 달려가려고 한다. 유섭이의 챔피언은 총  $N$  개의 분기점에 위치할 수 있다. 0번째 분기점은 현재 유섭이의 챔피언이 있는 곳을,  $N - 1$  번째 분기점은 상대편 넥서스를 의미하며 나머지  $1, 2, \dots, N - 2$  번째 분기점은 중간 거점들이다. 그러나 유섭이의 챔피언이 모든 분기점을 지나칠 수 있는 것은 아니다. 백도어의 핵심은 안 들키고 살금살금 가는 것이기 때문에 적 챔피언 혹은 적 와드(시야를 밝혀주는 토탈), 미니언, 포탑 등 상대의 시야에 걸리는 곳은 지나칠 수 없다.

입력으로 각 분기점을 지나칠 수 있는지에 대한 여부와 각 분기점에서 다른 분기점으로 가는데 걸리는 시간이 주어졌을 때, 유섭이가 현재 위치에서 넥서스까지 갈 수 있는 최소 시간을 구하여라.

### 입력 형식

첫 번째 줄에 분기점의 수와 분기점들을 잇는 길의 수를 의미하는 두 자연수  $N$  과  $M$  이 공백으로 구분되어 주어진다. ( $1 \leq N \leq 100,000$ ,  $1 \leq M \leq 300,000$ )

두 번째 줄에 각 분기점이 적의 시야에 보이는지를 의미하는  $N$  개의 정수  $a_0, a_1, \dots, a_{N-1}$  가 공백으로 구분되어 주어진다.  $a_i$  가 0이면  $i$  번째 분기점이 상대의 시야에 보이지 않는다는 뜻이며, 1이면 보인다는 뜻이다. 추가적으로  $a_0 = 0$ ,  $a_{N-1} = 1$  이다.  $N - 1$  번째 분기점은 상대 넥서스이기 때문에 어쩔 수 없이 상대의 시야에 보이게 되며, 또 유일하게 상대 시야에 보이면서 갈 수 있는 곳이다.

다음  $M$  개의 줄에 걸쳐 세 정수  $a, b, t$  가 공백으로 구분되어 주어진다. ( $0 \leq a, b < N$ ,  $a \neq b$ ,  $1 \leq t \leq 100,000$ ) 이는  $a$  번째 분기점과  $b$  번째 분기점 사이를 지나는데  $t$  만큼의 시간이 걸리는 것을 의미한다. 연결은 양방향이며, 한 분기점에서 다른 분기점으로 가는 간선은 최대 1개 존재한다.

### 출력 형식

첫 번째 줄에 유섭이의 챔피언이 상대 넥서스까지 안 들키고 가는데 걸리는 최소 시간을 출력하여라. 만약 상대 넥서스까지 갈 수 없으면 -1을 출력하여라.



**예제**

| 표준 입력   | 표준 출력 |
|---|-------|
| 5 7<br>0 0 0 1 1<br>0 1 7<br>0 2 2<br>1 2 4<br>1 3 3<br>1 4 6<br>2 3 2<br>3 4 1 | 12    |
| 5 7<br>0 1 0 1 1<br>0 1 7<br>0 2 2<br>1 2 4<br>1 3 3<br>1 4 6<br>2 3 2<br>3 4 1 | -1    |

**설명**

첫 번째 예제에서 위 그래프의 최단거리는  $0 \rightarrow 2 \rightarrow 3 \rightarrow 4$  를 지나는 시간인  $5(2+2+1)$  이지만, 3번 분기점이 상대의 시야에 있기 때문에  $0 \rightarrow 2 \rightarrow 1 \rightarrow 4$ 를 지나는 시간인  $12(2+4+6)$ 이 최소 시간이 된다.

## 문제 E. FLEX

시간 제한: 2 초

메모리 제한: 512 MB

요즘 스트레스를 많이 받는 인호에게 돈은 스트레스 해소 그 자체이다. 그 때문인지 인호는 어제 돈을 썼던만큼 오늘 쓰지 못하면, 만 원 단위로 표현한 차액의 제곱만큼 박탈감을 느낀다. 보다 엄밀히 서술하자면 다음과 같다.

만약  $i$  번째 날 지출이  $a$  만 원,  $(i + 1)$  번째 날 지출이  $b$  만 원이었다면,  $(i + 1)$  번째 날에

1.  $a \leq b$  라면 아무런 박탈감을 느끼지 않는다.
2.  $a > b$  라면  $(a - b)^2$  만큼의 박탈감을 느낀다.

인호는 앞으로  $N$  일 동안의 예상 통장 지출 내역을 살펴보고 있다. 인호의 통장에는 예상 지출과 별개로 여유롭게 사용할 수 있는  $M$  만 원의 돈이 남아있다. 인호는 이  $M$  만 원을 앞으로의 지출 내역에 효율적으로 분배하여, 앞으로  $N$  일 동안 느낄 박탈감의 합(즉, 내일의 박탈감, 모레의 박탈감, ...,  $N$  일 후의 박탈감의 합)을 최소화하고자 한다. 이 통장 잔고의 최소 인출 단위는 만 원이기 때문에, 분배는 만 원 단위로만 할 수 있다. 슬하고 더운 날씨 속에서 인호가 박탈감을 덜 받을 수 있게 도와주자.

### 입력 형식

첫 번째 줄에는 정산할 날짜의 수와 통장 잔고(만 원 단위)를 의미하는 두 자연수  $N$  과  $M$  이 공백으로 구분되어 주어진다. ( $1 \leq N < 1000$ ,  $0 \leq M < 200$ )

두 번째 줄에는  $N$  일 동안 지출할 예상 비용을 의미하는  $N$  개의 정수  $C_1, C_2, \dots, C_N$  가 공백으로 구분되어 주어진다. ( $0 \leq C_i \leq 10$ )

$C_i$  는  $i$  번째 날에 지출할 비용이  $C_i$  만 원임을 의미한다. 0 번째 날에 지출한 비용은 0으로 간주한다.

### 출력 형식

첫 번째 줄에 통장 잔고를  $N$  일간의 예상 지출액에 분배 하여 인호가 앞으로의  $N$  일 동안 느낄 박탈감의 총합의 최솟값을 출력한다.

### 예제

| 표준 입력        | 표준 출력 |
|--------------|-------|
| 3 1<br>3 2 1 | 1     |

### 설명

인호는 3일동안 각각 3, 2, 1만 원을 지출할 예정이고, 추가로 사용 가능한 잔액 1만 원이 남아있다.

원래는 총 2만큼의 박탈감을 느껴야 하지만 마지막 날에 1만 원을 추가한다면, 예상 지출 비용이 [3, 2, 2]가 되어 1만큼의 박탈감만을 느끼게 해줄 수 있다 (둘째 날에만 1을 느낀다).

3일 동안만의 박탈감을 따지는 것이기 때문에 4번째 날의 비용을 상정할 필요가 없음에 유의해야 한다.

이와 별개로, 기존의 예상 지출 자체를 재분배할 수는 없다. 예를 들어 [3, 2, 1]을 [2, 2, 2]로 바꾸는 것은 허용되지 않는다.

## 참고 사항

flex는 본디 구부리다, (준비 운동으로) 몸을 풀다의 뜻을 가지고 있는 영단어인데, 운동하며 나오는 근육을 과시함을 뜻하는 관용구 'flex your muscles'로도 쓰인다.

최근에는 몸풀기로 근육을 자랑하는 게 아니라 돈 등의 물질적인 자원을 자랑하는 은어로 사용되고 있다.

## 문제 F. 통신망 분할

시간 제한: 1 초

메모리 제한: 512 MB

BOJ의 인기스타, 방송인 권옥제는 통신 회사에 취업했다. 현재 이 통신 회사는 너무나 큰 통신망을 한 지사에서 관리하느라 큰 비용을 지불하고 있었다. 그래서 회사는 최근 IT의 트렌드 중 하나인 ‘탈중앙화’에 편승하여, 통신망을 분할하도록 결정했다. 그래서 옥제한테 통신망을 분할 할때 발생하는 비용을 분석하도록 지시했다.

현재 회사 망에는 1번부터  $N$ 번까지 총  $N$ 개의 통신 탑이 존재하며, 통신탑 간의 연결이  $M$ 개 존재한다. 이때 회사에서는 총  $Q$ 번 통신탑 간의 연결을 제거함으로써 하나의 통신망을 여러 개의 통신망으로 분리하려고 한다. 통신망이란, 통신탑의 연결을 통해 도달 가능한 통신탑들의 집합이다. 통신탑 간의 연결 관계를 제거할 때 드는 비용은 제거한 후 통신망이 두 개로 나누어진다면 나뉜 두 개의 통신망에 속한 통신탑들의 갯수의 곱이 되며, 나누어지지 않을 경우 0이다.

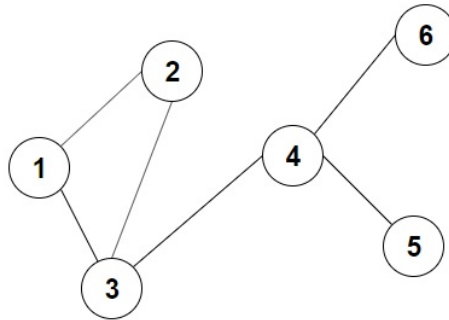


Figure 2: 예시 그래프

위의 그림을 예시로 할때, (3,4) 간의 연결을 제거하면 {1, 2, 3}, {4, 5, 6}으로 분할 되며, 이때 발생하는 비용은  $3 \times 3 = 9$ 가 된다. 대신 (2,3) 간의 연결을 제거하면, 망이 나뉘지지 않았기에 비용은 0이 된다.

옥제는 회사의 요청에 따라  $Q$ 번의 제거를 통해 나오는 비용의 합을 구해야 한다. 하지만 옥제는 회사의 통신망을 이용해 방송하기 바쁘기 때문에 우리가 도와주자.

### 입력 형식

첫 번째 줄에는 통신탑의 개수인 자연수  $N$ , 통신탑 사이의 연결의 개수인 자연수  $M$ , 통신망 연결 분할 횟수인 자연수  $Q$ 가 공백으로 구분되어 주어진다. ( $1 \leq N \leq 100,000$ ,  $1 \leq M \leq 100,000$ ,  $1 \leq Q \leq M$ )

두 번째 줄부터  $M$ 개의 줄에 걸쳐 두 개의 자연수  $X, Y$ 가 공백으로 구분되어 주어진다. 이는  $X$  통신탑과  $Y$  통신탑 사이에 연결이 있음을 뜻한다. ( $1 \leq X, Y \leq N$ ,  $X \neq Y$ )

중복된 연결은 주어지지 않으며, 모든 통신탑은 처음엔 하나의 통신망에 속한다. 조건에 의해 자기 자신과 연결이 있는 통신탑은 없다.

그 다음 줄부터  $Q$ 개의 줄에 걸쳐 제거될 연결의 번호인 자연수  $A$ 가 주어진다. 이는  $A$ 번째로 입력된  $(X, Y)$ 의 연결이 제거되었음을 의미한다. ( $1 \leq A \leq M$ )

이미 제거된 연결은 다시 제거되지 않으며, 제거는 입력 순서대로 진행된다.

## 출력 형식

첫 번째 줄에  $Q$ 개의 연결을 순서대로 제거하는데 드는 비용의 합을 출력한다.

## 예제

| 표준 입력  | 표준 출력 |
|--|-------|
| 4 4 3<br>1 2<br>2 3<br>3 4<br>1 4<br>4<br>2<br>3 | 5     |

## 설명

첫 번째로 제거되는 연결은 (1,4)로, 통신망 {1, 2, 3, 4}가 분리되지 않아 비용이 0이다.

두 번째로 제거되는 연결은 (2,3)으로, 통신망 {1, 2, 3, 4}가 {1, 2}와 {3, 4}로 분리되어 비용이  $2 \times 2 = 4$ 이다.

세 번째로 제거되는 연결은 (3,4)로, 통신망 {3, 4}가 {3}과 {4}로 분리되어 비용이  $1 \times 1 = 1$ 이다.

결과적으로 총 비용은  $0 + 4 + 1 = 5$ 이다.

## 문제 G. 트리의 외심

시간 제한: 2 초

메모리 제한: 512 MB

알고리즘에 푼 빠진 동판이가 트리에 심취한 나머지 트리에서 외심을 정의하려 한다. 트리란, 모든 정점이 연결되어 있으면서 사이클이 존재하지 않는 그래프이다. 하지만 동판이는 트리에서 외심을 정의하기 위해서는 “트리에서 두 정점 사이의 거리”도 정의해야 한다는 사실을 깨달았다!

트리에서 두 정점 사이의 거리는 한 정점에서 다른 정점으로 가기 위해 거쳐야 하는 최소한의 간선의 개수로 정의된다. 이 때 트리의 세 정점에 대해, 트리의 외심은 세 정점으로부터 거리가 같으면서, 그 거리를 최소로 하는 정점이 존재한다면 해당 정점으로 정의된다. 수학적으로 트리의 세 정점에 대해 외심이 존재한다면, 유일하다는 것을 보일 수 있다.

자명하게도, 외심을 정의하는 3개의 정점이 달라지면 같은 트리라 해도 외심이 달라진다. 동판이는 다양한 외심들을 찾아보고 싶지만 코딩에 귀찮음을 겪고 있다.....동판이를 위해 여러분들이 대신 코드를 짜주도록 하자.

### 입력 형식

첫 번째 줄에 정점의 개수  $N$ 이 주어진다. ( $1 \leq N \leq 100,000$ ) 이 트리는 1번 정점, 2번 정점, ...,  $N$ 번 정점으로 구성된다.

두 번째 줄부터  $N$ 번째 줄까지, 트리의 간선 정보를 의미하는 두 자연수  $X, Y$ 가 공백으로 구분되어 주어진다. 이는  $X$ 번 정점과  $Y$ 번 정점이 연결되어있음을 의미한다. ( $1 \leq X, Y \leq N, X \neq Y$ )

주어지는 연결관계는 트리를 구성한다.

$N + 1$  번째 줄에는 쿼리의 개수  $Q$ 가 주어진다. ( $1 \leq Q \leq 100,000$ )

다음  $Q$ 개의 줄에 걸쳐, 외심을 정의하기 위한 세 개의 정점 번호를 뜻하는 세 자연수  $A, B, C$ 가 공백으로 구분되어 주어진다. ( $1 \leq A, B, C \leq N$ )

### 출력 형식

$Q$ 개의 줄에 걸쳐 각 쿼리마다 입력으로 주어진 세 정점에 대해 트리의 외심이 존재하면 외심의 정점 번호를, 존재하지 않으면 -1을 출력한다.

**예제**

| 표준 입력  | 표준 출력   |
|--|---------|
| 4<br>1 2<br>1 3<br>1 4<br>2<br>2 3 4<br>1 2 3      | 1<br>-1 |
| 2<br>1 2<br>2<br>1 1 2<br>2 2 2                    | -1<br>2 |
| 6<br>1 2<br>2 3<br>2 4<br>3 5<br>5 6<br>1<br>1 4 6 | 3       |