



</>

2020

SKH
Algorithm
Contest

Intermediate

대회가 시작되기 전까지 절대 표지를 넘기지
마세요.



고려대학교
SW중심대학사업단



숭실대학교
스파르탄SW교육원

NAVER D2

wishket

대회 규칙

- 사용 가능한 언어는 **C, C++, Java, Python 3, PyPy 3**입니다.
 - 모든 문제에 대해 제약 조건을 만족하며 정답을 출력하는 C++17 코드가 있음이 보장됩니다.
- 대회는 대회 전용 DOMjudge 사이트에서 치뤄지며 문제, 채점 실시간 정보 등을 확인할 수 있습니다.
- 순위는 푼 문제가 많은 순서대로, 푼 문제 수가 같을 경우에는 패널티의 합이 낮은 순으로 정렬됩니다.
 - Contest(Intermediate/Advanced)에서 문제별 패널티는 '(문제를 풀기까지 걸린 시간(분)) + (그 전까지 제출한 횟수) × 10'입니다.
 - 컴파일 에러는 제출 횟수에 포함되지 않습니다.

금지 / 제한 행위

- 대회 대리 참가를 금합니다.
- Contest(Intermediate/Advanced)의 경우 참가자의 정면이 보이는 캠 공유는 대회가 끝날 때까지 중단되어서는 안 됩니다.
- Marathon의 경우 캠 공유가 꺼져있는 상태에서의 답안 제출은 금합니다. (24시간 내내 공유하지 않아도 됩니다.)
- 컴퓨터를 두 대 이상 사용하는 것을 금합니다.
- 운영진에게 질문하는 것 외에 다른 사람과 대화하는 것을 금합니다.
- 허용된 레퍼런스 페이지를 제외한 메신저, 대화, 이동식 저장 매체를 통한 문제 풀이를 금합니다.
- Contest(Intermediate/Advanced)의 경우 인터넷 검색을 통한 문제 풀이를 금합니다.
- 문제 제출을 비정상적으로 많이 시도하거나, 의도적으로 대회 웹 서버를 공격하는 행위를 금합니다.

대회 규칙을 어기거나, 운영진이 판단하기에 부정한 행위를 저지를 경우 경고 없이 대회 참가 자격이 박탈될 수 있습니다.

레퍼런스 사이트

다음 레퍼런스 사이트는 열람할 수 있습니다.

- **C/C++** : <https://en.cppreference.com/w/>, <http://cplusplus.com>
- **Java** : <https://docs.oracle.com/en/java/javase/11/docs/api/index.html>
- **Python** : <https://docs.python.org/3/>

채점 서버 환경

채점 서버의 플랫폼은 **DOMjudge**이며, 운영체제는 **Ubuntu 7.5.0-3ubuntu1 18.04 (linux kernel 4.15.0-106-generic)**입니다.

컴파일 옵션 및 시간 보정

사용 가능한 언어와 컴파일 옵션 및 제한 시간 보정은 다음과 같습니다. "\$@"는 업로드한 코드 및 생성된 프로그램 이름입니다.

- **C14** (gcc 8.3.0)

컴파일 : gcc -x c -Wall -O2 -std=c14 -static -pipe -o "\$DEST" "\$@" -lm
실행 : exec "\$@"

- **C++17** (g++ 8.3.0)

컴파일 : g++ -x c++ -Wall -O2 -std=c++17 -static -pipe -o "\$DEST" "\$@"
실행 : exec "\$@"

- **Java** (Java 11.0.4)

컴파일 : javac -encoding UTF-8 -sourcepath . -d . "\$@"
실행 : exec java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss\${MEMSTACK}k \ -Xms\${MEMLIMITJAVA}k -Xmx\${MEMLIMITJAVA}k '\$MAINCLASS' '\\$@'

– MEMSTACK은 65536이며, MEMLIMIT은 문제의 제한에서 128MB를 뺀 값입니다.

시간 : 3배

- **Python 3** (Python 3.7.3)

컴파일 : python3 -m py_compile "\$@"

실행 : python3 "\$@"

시간 : 4배

- **PyPy 3** (PyPy 7.0.0 with GCC 8.2.0 (Python 3.5.3))

컴파일 : python3 -m py_compile "\$@"

실행 : pypy3 "\$@"

시간 : 4배

DOMjudge 채점

DOMjudge에 코드를 업로드할 때는 다음 조건을 지켜야 합니다.

- 파일 이름은 알파벳 및 숫자로 시작해야 하며, 알파벳 대소문자 / 숫자 / +_._-만 사용 가능합니다.
- 확장자는 C는 .c, C++는 .cpp / cc / cxx / c++, Java는 .java, Python은 .py / .py3여야 합니다.
- 제출한 코드는 표준 입출력만으로 통신하여야 합니다 (파일 입출력은 금지됩니다).
- 제출한 소스코드의 크기는 256 MiB 이하여야 합니다.
- 한 문제에 제출은 최대 100번 할 수 있습니다.

DOMjudge 채점 결과

Submit을 한 다음에 Scoreboard 탭에서 제출 결과를 확인할 수 있습니다.

- PENDING : 제출되었으며, 채점 대기중이거나 채점중입니다.

- **CORRECT** : 제출한 코드가 모든 테스트 케이스에 대해 시간 제한 / 메모리 제한 내에서 올바른 답을 내었고, 정상적으로 종료되었습니다. 이 경우 제출자는 해당 문제를 풀었습니다.
- **COMPILER-ERROR** : 컴파일 과정 중에 에러가 발생하여 채점이 진행되지 않았습니다.
- **TIMELIMIT** : 프로그램 수행 시간이 제한 시간을 초과하였습니다.
- **RUN-ERROR** : 프로그램 수행 중 에러가 발생하였습니다. (예시 : 0으로 나누기, 잘못된 주소 참조)
- **WRONG-ANSWER** : 프로그램이 오답을 출력하였습니다.
- **OUTPUT-LIMIT** : 프로그램이 지나치게 많은 출력을 하였습니다.

CORRECT가 아닌 결과가 나온 테스트 케이스가 최초로 등장하면, 해당 채점 결과가 제출 결과로 보여집니다.
대회 중 ‘request clarification’ 탭을 통해 주최진에게 질문을 물을 수 있습니다.

문제 목록

대회는 4시간 동안 진행되며, 총 9문제로 구성되어 있습니다.

총 문제지가 표지를 제외하고 17쪽인지 확인하시길 바랍니다.

문제의 목록은 다음과 같으며, 출제진이 판단하기에 난이도 순으로 정렬되어 있습니다.

- A SKH 엔터테인먼트 (skhent)
- B 균형잡힌 수열 (balanced)
- C 나무를 심어볼까요? (planting)
- D 제가 재밌는 문제를 가져왔어요 (fun)
- E 제곱 가중치 (weight)
- F 전염병을 피해라 (virus)
- G 강남 건물주 (gangnam)
- H 최소 대역폭 보장 (bandwidth)
- I 오렌지 수거 (orange)

문제 A. SKH 엔터테인먼트 (skhent)

시간 제한: 1 초
메모리 제한: 512 MB

세계적인 스타들을 배출한 SKH 엔터테인먼트는 1년 동안 팬들에게서 오는 선물들을 소속된 가수들의 다음 해 생일에 맞춰 몰아서 주곤 한다. 여기에 사용할 상자는 굉장히 많은 선물을 담을 수 있는 특이한 상자여야 하므로, 주문제작을 위해 매달 몇 개의 상자가 필요할지 알아낼 필요가 있다. 상자에 선물을 담을 때, 다음과 같은 규칙을 따른다고 하자.

1. 한 사람에게 가는 선물들은 하나의 상자에 모두 담는다.
2. 각 선물에는 스타의 이름과 생일이 적혀있다.
3. 동명이인은 없다.
4. 한 사람에게 여러 개의 선물이 올 수 있다.
5. 편의를 위해 모든 달은 31일까지 있다고 가정한다.

입력 형식

첫 번째 줄에는 선물의 개수 N 이 주어진다. ($1 \leq N \leq 100,000$)

두 번째 줄부터 N 개의 줄에 걸쳐 생일과 이름이 'MM DD 이름'의 형식으로 주어진다. 이름은 20자 이하의 알파벳 대문자로 이루어져 있으며, 공백문자는 포함되지 않는다.

출력 형식

열두 개의 줄에 걸쳐 i 번째 줄에 i 월에 몇 개의 상자가 필요한지 출력한다. ($1 \leq i \leq 12$)

예제

표준 입력	표준 출력
10	2
05 28 DAHYUN	0
05 16 IU	1
03 29 IRENE	0
09 22 NAYEON	2
12 29 SANA	1
06 14 TZUYU	0
01 03 JISOO	0
01 16 JENNIE	1
10 10 SUZY	1
05 28 DAHYUN	0
	1

참고 사항

예제 입출력

문제 B. 균형잡힌 수열 (balanced)

시간 제한: 1초
메모리 제한: 512 MB

길이 N 인 수열 A 가 주어진다.

끝 부분에 위치하지 않은 모든 원소에 대해 자신과 인접한 원소들의 평균값이 자신의 값과 같은 경우 이 수열을 “균형잡힌 수열”이라고 한다. 다시 말해, $2 \leq i \leq N - 1$ 에 대해 $\frac{A_{i-1} + A_{i+1}}{2} = A_i$ 를 만족해야 한다.

A 를 균형잡힌 수열이 되도록 순서를 재배열할 수 있는지 판정하고, 가능하다면 재배열된 수열을 출력하라.

입력 형식

첫 번째 줄에는 N 이 주어진다. ($3 \leq N \leq 2000$)

두 번째 줄에는 수열 A 원소 A_1, A_2, \dots, A_N 이 주어진다. ($-10^9 \leq A_i \leq 10^9$)

입력으로 들어오는 수는 모두 정수임이 보장된다.

출력 형식

A 를 균형잡힌 수열이 되도록 재배열할 수 없다면 첫 번째 줄에 “No”(따옴표 제외)를 출력한다.

재배열하는 방법이 존재한다면 첫 번째 줄에 “Yes”(따옴표 제외)를 출력하고, 두 번째 줄에 수열 A 의 재배열된 원소 N 개를 차례대로 공백으로 구분하여 출력한다.

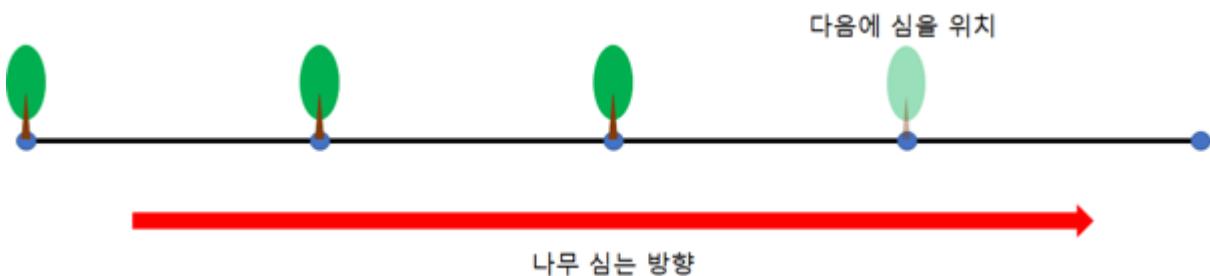
예제

표준 입력	표준 출력
3 3 1 5	Yes 5 3 1
6 1 4 2 8 5 7	No

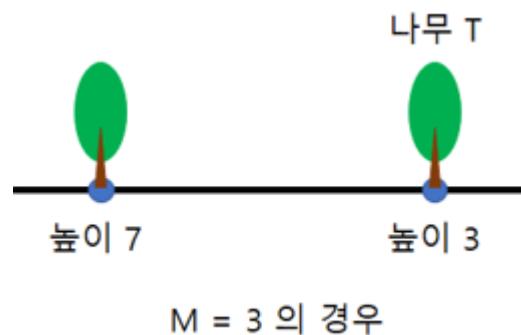
문제 C. 나무를 심어볼까요? (planting)

시간 제한: 1초
메모리 제한: 512 MB

정호는 송고한 정원에 나무를 매일 한 그루씩 심고 있다. 정원을 아래와 같이 일정한 거리로 나눠서 왼쪽에서부터 오른쪽으로 한 칸씩 나무를 심으려 한다.

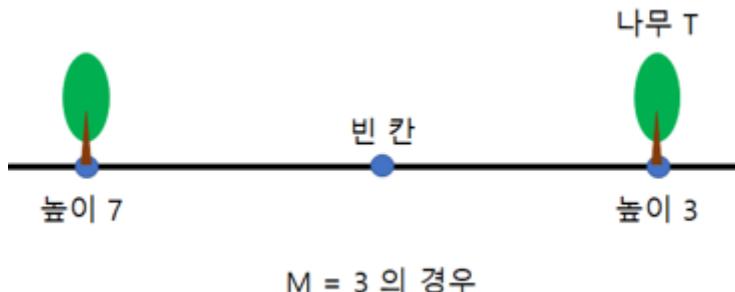


그런데 여기서 하나의 문제가 생겼다. 아래와 같이 어떤 나무를 T 라고 할 때, T 의 바로 왼쪽의 나무의 높이가 T 의 높이보다 M 이상 높다면, T 는 왼쪽의 그늘에 가려져서 햇빛을 받을 수 없는 것이다.



$M = 3$ 의 경우

물론 아래와 같이 한 칸 띄워서 나무를 심게 되면 T 도 햇빛을 받을 수 있다.



$M = 3$ 의 경우

가장 일을 쉽게 하는 방법은 지금 심는 나무가 햇빛을 못 받을 조건이라면 그냥 한 칸 띄워서 심는 것이다. 하지만, 이 경우 정원 곳곳에 빈 칸이 생기기 때문에 미관상 좋지 않다. 그래서 정호는 다음과 같은 규칙으로 나무를 심으려 한다.

1. 나무 심기는 왼쪽에서 오른쪽 방향으로 진행한다.
2. 비어있는 칸 중 가장 왼쪽 칸에 나무를 심었을 때 이 나무가 햇빛을 받을 수 있다면, 바로 나무를 심는다.
3. 만약 나무를 심었을 때 이 나무가 햇빛을 받을 수 없다면, 조건에 따라 다음 중 하나를 수행한다.
 - 현재 심어져 있는 나무 중 가장 오른쪽의 나무가 아직 심어진 지 K 일이 되지 않았다면, 그 나무를 뽑아서 버린 후, 1의 과정으로 돌아간다.
 - 현재 심어져 있는 나무 중 가장 오른쪽의 나무가 심어진 지 K 일 이상이 되었다면, 이미 뿌리가 깊게 박혀서 뽑기 힘들기 때문에 한 칸 띄워서 나무를 심는다. 이때 생기는 빈 칸에는 다시 나무를 심을 수 없다.

이런 방식으로 나무를 심었을 때, 과연 정원에 빈 칸은 총 몇 개가 생길까?

입력 형식

첫 번째 줄에 나무의 그루 수 N 과, M , K 이 공백으로 구분되어 주어진다. ($1 \leq N, K \leq 100,000$, $1 \leq M \leq 10,000$)

두 번째 줄부터 N 개의 줄에 걸쳐 i 번째 날의 나무의 높이 H_i 가 주어진다. ($1 \leq H_i \leq 10,000$)

출력 형식

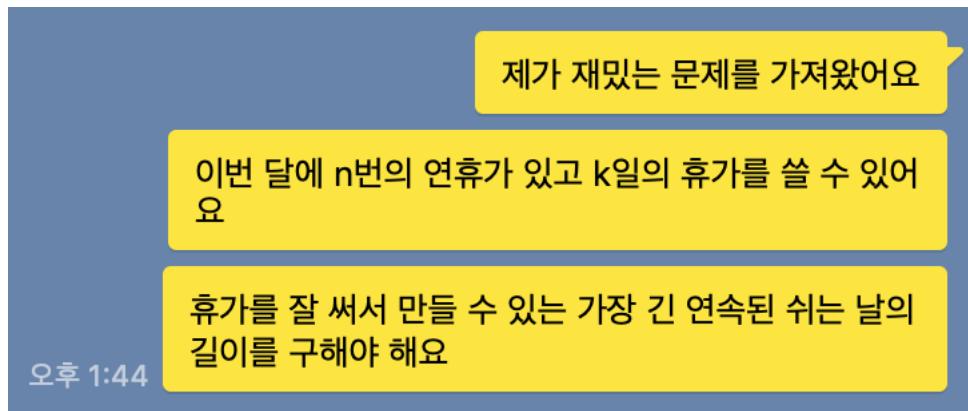
정호가 나무를 심었을 때 생기는 빈 칸의 개수를 출력한다.

예제

표준 입력	표준 출력
8 2 2	2
7	
6	
4	
5	
3	
8	
6	
1	

문제 D. 제가 재밌는 문제를 가져왔어요 (fun)

시간 제한: 1초
메모리 제한: 512 MB



입력 형식

첫 번째 줄에 연휴의 개수 N , 휴가의 개수 K 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 200,000$, $1 \leq K \leq 1,000,000$)

두 번째 줄부터 N 개의 줄에 걸쳐, 연휴의 시작일 s_i 과 끝일 e_i 이 공백으로 구분되어 주어진다. ($1 \leq s_i \leq e_i \leq 1,000,000$)

임의로 2개의 연휴를 골랐을 때, 교집합이 존재하는 경우는 없다.

출력 형식

휴가를 적절히 사용해서 만들 수 있는 가장 긴 연속된 쉬는 날의 길이를 출력한다.

예제

표준 입력	표준 출력
5 3 1 3 5 6 9 10 14 16 22 27	10
5 3 1 3 5 6 9 10 14 16 22 33	15
4 1 2 3 5 6 12 13 15 17	6

문제 E. 제곱 가중치 (weight)

시간 제한: 1초
메모리 제한: 256 MB

N 개의 정수 a_1, a_2, \dots, a_N 이 주어진다.

$1^2 \cdot a_1, 2^2 \cdot a_1 + 1^2 \cdot a_2, \dots, N^2 \cdot a_1 + (N-1)^2 \cdot a_2 + \dots + 1^2 \cdot a_N$ 을 각각 출력한다.

바꿔 말하면, $k = 1, 2, \dots, N$ 에 대해서 $\sum_{i=0}^{k-1} (k-i)^2 \cdot a_{i+1}$ 를 출력한다.

입력 형식

첫 번째 줄에 정수의 개수 N 이 주어진다. ($1 \leq N \leq 100,000$)

두 번째 줄에 N 개의 정수 a_1, a_2, \dots, a_N 이 공백으로 구분되어 주어진다. ($-1,000 \leq a_i \leq 1,000$)

출력 형식

위의 계산으로 얻을 수 있는 N 개의 수들을 공백으로 구분하여 출력한다.

예제

표준 입력	표준 출력
4 1 2 3 4	1 6 20 50

참고 사항

첫 번째 예시에서, $a = (1, 2, 3, 4)$ 이므로, $1^2 \cdot a_1 = 1$, $2^2 \cdot a_1 + 1^2 \cdot a_2 = 6$, $3^2 \cdot a_1 + 2^2 \cdot a_2 + 1^2 \cdot a_3 = 20$, 그리고 $4^2 \cdot a_1 + 3^2 \cdot a_2 + 2^2 \cdot a_3 + 1^2 \cdot a_4 = 50$ 이다.

문제 F. 전염병을 피해라 (virus)

시간 제한: 1초
메모리 제한: 256 MB

동관이가 사는 나라에서 전염병이 발생하였다. 따라서 동관이는 전염병을 피해서 병원이 있는 도시로 가려고 한다.

전염병은 여러 지역에서 발생하였으며 도로를 통해 퍼져나간다. 만약 전염병이 어떤 도시에 도착한다면, 그 도시는 사람이 갈 수 없도록 봉쇄된다. 즉, 동관이는 전염병이 아직 도착하지 않은 도시를 통해서 병원으로 가야 하며, 도시가 전염됨과 동시에 도시에 도착하는 상황도 피해야 한다. 단, 병원이 있는 도시는 전염병에 감염되지 않는다고 가정한다.

도시와 도시 사이는 양방향 도로로 연결되며, 도로에는 가중치가 존재한다. 간선에 있는 가중치는 전염병이나 동관이가 그 간선을 타고 다른 도시로 가는데 걸리는 시간을 의미한다.

동관이가 사는 나라의 정보가 그래프로 주어지면, 동관이가 병원에 가는 데 걸리는 최소 시간을 구하여라.

입력 형식

첫 번째 줄에 도시의 개수 N , 도로의 개수 M , 전염병 진원지의 수 U 가 공백으로 구분되어 주어진다.
($2 \leq N \leq 100,000$, $N - 1 \leq M \leq 300,000$, $0 \leq U \leq N - 2$)

M 개의 줄에 걸쳐 도로가 연결하는 두 도시 x, y 와 가중치 c 가 공백으로 구분되어 주어진다.
($1 \leq x, y \leq N$, $x \neq y$, $1 \leq c \leq 1,000,000$)

$M + 2$ 번째 줄에 서로 다른 U 개의 전염병 진원지 정보 z_i 가 공백으로 구분되어 주어진다.
($2 \leq z_i \leq N$, $1 \leq i \leq U$)

$M + 3$ 번째 줄에 병원의 정보 h 이 주어진다. ($2 \leq h \leq N$)

동관이의 초기 위치는 1이며, 입력으로 주어지는 그래프는 연결그래프임이 보장된다.

출력 형식

동관이가 병원에 가는데 걸리는 최소 시간을 출력한다. 만약 동관이가 병원에 갈 수 없다면 -1을 출력한다.

예제

표준 입력	표준 출력
5 5 1 1 2 3 1 3 3 2 5 1 1 4 100 4 3 1 4 5	4
7 9 2 1 2 7 2 3 4 3 4 10 1 4 9 3 7 20 4 7 70 4 5 5 5 6 5 6 7 100 2 6 7	79
5 6 1 1 2 3 1 3 3 2 5 1 1 4 100 4 3 1 3 2 2 4 5	-1

문제 G. 강남 건물주 (gangnam)

시간 제한: 1초
메모리 제한: 512 MB

일평생 꿈꿔왔던 건물주의 꿈을 드디어 이루게 된 경수는, 강남에 N 층짜리 건물을 짓게 되었다. 자신의 꿈을 이뤄줄 건물이므로, 경수는 특별한 건물을 만들기 위해, 아주 특별한 엘리베이터를 설치하게 되었다.

이 엘리베이터는 버튼이 2개 밖에 없는데, A 버튼을 누르면 지금 있는 곳에서 한 층을 올라가고, B 버튼을 누르면 방금 올라온 층 수의 2배만큼 올라간다. 예를 들어, 1층에 있는 상태에서 $A \rightarrow B \rightarrow B \rightarrow A \rightarrow B$ 의 순서로 버튼을 누르면, 1층 \rightarrow 2층 \rightarrow 4층 \rightarrow 8층 \rightarrow 9층 \rightarrow 11층에 도달하게 된다.

경수는 이 건물에 대한 입주공고를 M 번 내는데, 각 입주 공고는 (P 층, Q 층)의 형태로 나온다. 이는 P 층 이상 Q 층 이하의 층들에 입주가 가능하다는 뜻이다. 이 소식을 들은 건이는 입주공고가 나올 때마다, 해당 범위의 층에 들어가기 위해서는, 1층에서 최소한 몇 번의 버튼을 눌러야 하는지 궁금해졌다. 입주 공고의 층수의 범위가 주어질 때, 1층에서 해당 범위 안의 층에 도달하기 위해 최소 몇 번의 버튼을 눌러야 하는지 구해보자.

입력 형식

첫 번째 줄에 건물의 층수 N 이 주어진다. ($1 \leq N \leq 100,000$)

두 번째 줄에 입주 공고가 나올 횟수 M 이 주어진다. ($1 \leq M \leq 100,000$)

세 번째 줄부터 M 개의 줄에 걸쳐 입주할 수 있는 층수의 범위 P_i, Q_i 가 공백으로 구분되어 주어진다. ($1 \leq P_i \leq Q_i \leq N$)

출력 형식

M 개의 줄에 걸쳐 i 번째 줄에 i 번째 입주 공고에서 1층에서 P_i 층 이상 Q_i 층 이하의 층들에 도달하기 위해 최소 몇 번의 버튼을 눌러야 하는지를 출력한다. ($1 \leq i \leq M$)

예제

표준 입력	표준 출력
4	0
3	2
1 2	1
3 4	
2 4	

문제 H. 최소 대역폭 보장 (bandwidth)

시간 제한: 3초
메모리 제한: 256 MB

당신은 인터넷 서비스 업체에서 일하는 직원이다. 인터넷 업체에서는 여러 도시 사이를 간선으로 연결하고 있으며, 각 간선에는 대역폭이라는 개념이 있다. 여기서 대역폭은 단위 시간 동안 얼마나 많은 양의 데이터가 그 간선을 통해서 흘러갈 수 있는지를 의미한다.

당신이 담당하고 있는 여러 도시에는 데이터가 흘러가는 통신들이 존재한다. 당신은 현재의 인터넷망이 적절한지 확인해 달라는 의뢰를 받았다.

각 통신이 이루어지는 시간은 서로 다르기에 각 통신은 서로 연관되어 있지 않은 개별적인 것으로 보아도 무방하다. 즉, 모든 통신은 다른 통신의 대역폭에 영향을 끼치지 않는다.

질의는 도시 A 와 도시 B 를 최소 대역폭 C 로 이어줄 수 있는 경로가 존재하는지의 여부를 묻게 된다. 경로를 구성하는 각 간선은 대역폭 C 를 감당할 수 있어야만 한다. 이러한 경로가 있다면 “Yes”(따옴표 제외), 없다면 “No”(따옴표 제외)를 출력하여라.

입력 형식

첫 번째 줄에 도시의 개수인 N , 간선의 개수 M , 질의의 개수 Q 가 공백으로 구분되어 주어진다. (단, $3 \leq N \leq 100,000$, $N - 1 \leq M \leq 600,000$, $0 \leq Q \leq 300,000$)

두 번째 줄부터 M 개의 줄에 걸쳐 도시를 잇는 간선의 정보가 주어진다. 간선을 이어주는 두 개의 도시인 C_1 , C_2 와 간선의 가중치 V_1 이 공백으로 구분되어 주어진다. ($1 \leq C_1, C_2 \leq N$, $C_1 \neq C_2$, $1 \leq V_1 \leq 10^8$)

$M+2$ 번째 줄부터 Q 개의 줄에 걸쳐 질의가 주어진다. A, B, C 순서대로 숫자가 공백으로 구분되어 주어진다. ($1 \leq A, B \leq N$, $A \neq B$, $1 \leq C \leq 10^8$).

인터넷 망은 연결그래프임이 보장되며, 도시와 도시를 직접 연결하는 간선이 여러 개일 경우 통신에는 최대한 개의 간선만 포함될 수 있음에 유의하여라.

출력 형식

Q 개의 줄에 걸쳐 i 번째 질의에 대한 답 “Yes” / “No”(따옴표 제외)를 출력한다.

예제

표준 입력	표준 출력
6 7 5	Yes
1 2 10	Yes
2 4 5	No
5 1 9	No
3 2 17	Yes
4 5 60	
4 6 14	
3 6 19	
1 6 10	
3 4 13	
6 2 18	
2 4 16	
1 5 10	
9 18 12	Yes
3 6 24152544	Yes
4 1 11790068	No
6 4 53454943	Yes
3 4 63940950	No
6 1 94610065	Yes
4 8 57342012	No
1 6 54152544	No
8 3 54152544	Yes
2 5 24152544	No
7 6 28381629	No
9 7 23853037	Yes
6 8 63940950	
2 7 93632195	
5 8 3315776	
5 3 24152544	
1 7 22124296	
4 3 47111183	
2 8 21164502	
7 5 880086	
9 6 19591951	
3 8 78809055	
3 1 56785509	
9 5 67383541	
7 4 5632302	
8 1 75761682	
3 1 58443816	
6 4 19800905	
1 5 24352854	
7 3 99078163	
2 7 45375281	

문제 I. 오렌지 수거 (orange)

시간 제한: 5초
메모리 제한: 256 MB

Amel은 Orange가 너무 싫은 나머지, 온 몸이 보라색으로 변하는 저주에 걸렸다! 이 저주를 풀기 위하여, 숭고한 도시의 N 개의 마을에 존재하는 모든 오렌지를 없애버리기로 결심하였다.

마을에는 오렌지가 너무 많아, 전용 쓰레기통에 버려야만 한다. 전용 쓰레기통이 존재하는 마을은 정확히 두 군데이다. Amel은 모든 오렌지를 모아, 이 두 개의 쓰레기통에 넣을 것이다. 모든 오렌지에 대해 각각 두 쓰레기통 중 아무 쓰레기통에 넣으면 된다. (꼭 하나의 쓰레기통을 골라 모든 오렌지를 모아야 하는 것이 아니다.)

Amel은 오렌지가 싫기 때문에, 오렌지를 홀수 개만큼 들고 마을과 마을 사이를 다니는 것은 마을 사이의 거리 D 만큼의 힘이 든다. 그런데 짹수 개의 오렌지를 운반하는 데에는 힘이 하나도 들지 않는다고 한다. 또한 당연히 같은 마을 안에서 돌아다니는 것은 힘이 들지 않는다.

마을들은 트리의 형태로 이어져 있다(사이클이 존재하지 않는다). 또한 어떤 마을에서도 임의의 다른 마을로 가는 길이 존재한다. 쓰레기통이 있는 두 마을의 위치와 마을들을 연결하는 도로의 정보가 주어졌을 때, Amel이 최소한의 힘을 들여 모든 오렌지를 수거할 수 있는 방법을 찾아라.

입력 형식

첫 번째 줄에 숭고한 도시에 있는 마을의 수 N 과 쓰레기통이 있는 마을의 번호 A, B 가 공백으로 구분되어 주어진다. 마을은 1번부터 N 번까지 있다. ($2 \leq N \leq 1,000,000, 1 \leq A, B \leq N, A \neq B$)

두 번째 줄부터 $N - 1$ 개의 줄에 걸쳐, 각 도로의 양 끝 마을번호 s, e 와 거리 d 가 주어진다. ($1 \leq s, e \leq N, s \neq e, 0 \leq d \leq 10^9$)

N 번째 줄에 각 마을에 있는 오렌지의 개수 O 가 1번 마을부터 N 번 마을까지 순서대로 공백으로 구분되어 주어진다. ($0 \leq O \leq 10^9$)

모든 마을은 서로 연결되어 있고, 트리임이 보장된다.

출력 형식

Amel이 들여야 하는 최소한의 힘을 출력하여라.

예제

표준 입력	표준 출력
4 1 2 1 2 1 1 3 2 2 4 3 1 2 3 4	2
9 2 5 1 2 5 2 3 10 4 2 6 1 5 10 5 9 4 6 5 8 6 7 4 6 8 2 5 7 9 3 8 4 7 3 1	31