



</>

2020

SKH
Algorithm
Contest

Advanced

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 18쪽인지 확인하시길 바랍니다.

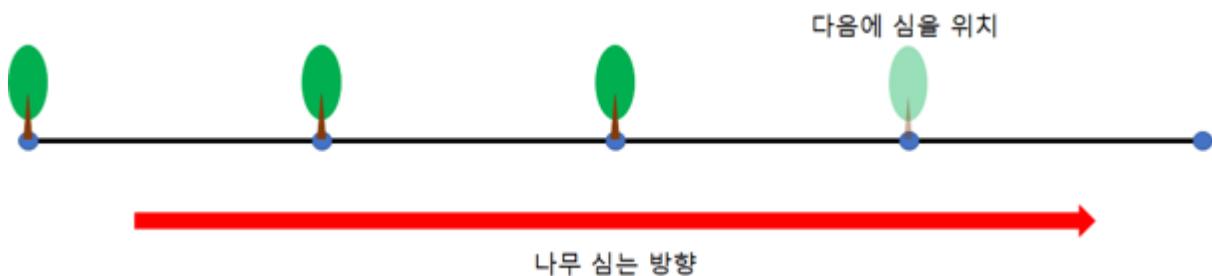
문제의 목록은 다음과 같으며, 출제진이 판단하기에 난이도 순으로 정렬되어 있습니다.

- A 나무를 심어볼까요? (planting)
- B 제곱 가중치 (weight)
- C 강남 건물주 (gangnam)
- D 최소 대역폭 보장 (bandwidth)
- E 동굴의 입구를 열어라! (cave)
- F 송고한 아이스크림 (icecream)
- G 수열 복원 (restore)
- H 반사복제된 트리 (mirrortree)
- I 찾아라 드래곤볼! (dragonball)

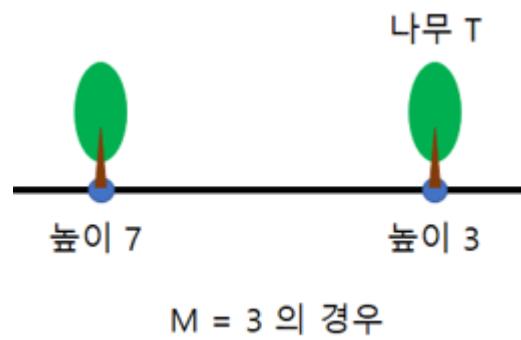
문제 A. 나무를 심어볼까요? (planting)

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.

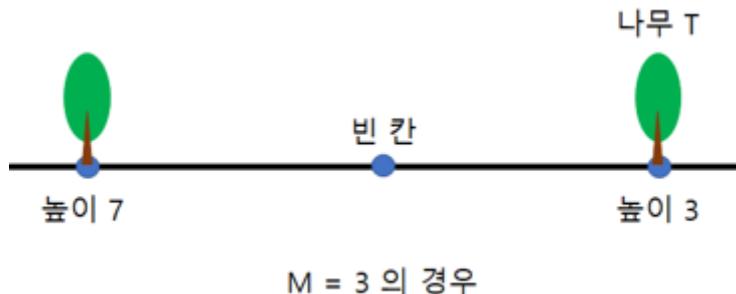


그런데 여기서 하나의 문제가 생겼다. 아래와 같이 어떤 나무를 T 라고 할 때, T 의 바로 왼쪽의 나무의 높이가 T 의 높이보다 M 이상 높다면, T 는 왼쪽의 그늘에 가려져서 햇빛을 받을 수 없는 것이다.



$M = 3$ 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 B. 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 C. 강남 건물주 (gangnam)

시간 제한: 1 초
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, A버튼을 누르면 지금 있는 곳에서 한 층을 올라가고, B버튼을 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 D. 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 E. 동굴의 입구를 열어라! (cave)

시간 제한: 4초
메모리 제한: 8MB

동관이는 동굴을 탐험하던 중, 비밀의 문을 발견했다. 그 문에는 다음과 같이 쓰여 있다 :

동굴에 있는 돌들을 가져와, 돌들에 쓰여 있는 숫자의 합이 정확히 M 이 되도록 하여라.

신기하게도, 동굴에 있는 돌은 가져와도 사라지지 않는다. (즉, 똑같은 돌을 여러 번 가져올 수 있다.)

그러나, 동관이는 움직이기 매우 귀찮다. 동관이를 위해 동굴에 있는 돌들을 최소 몇 개 가져와야 하는지, 또 그러한 쌍이 몇 개인지 알려주자!

일반적이지 않은 메모리 제한에 유의하여라.

입력 형식

첫 번째 줄에는 돌에 쓰여 있는 숫자들의 종류 N 이 주어진다.

두 번째 줄에는 돌에 쓰여 있는 서로 다른 N 개의 자연수 A_i 가 공백으로 구분되어 주어진다.

($2 \leq N, 1 \leq A_1 < A_2 < \dots < A_N \leq 10^7, A_2 \times A_3 \times \dots \times A_{N-1} \leq 3 \times 10^7$)

세 번째 줄에는 M 이 주어진다. ($1 \leq M \leq 10^{14}$)

출력 형식

동굴에서 가져와야 하는 돌의 개수의 최솟값을 출력한다.

또, 돌들에 쓰여 있는 숫자의 합이 M 이면서 돌의 개수가 최소가 되도록 하는 순서쌍 (x_1, x_2, \dots, x_n) 의 수를 공백으로 구분하여 출력한다. (x_i 는 동굴에서 a_i 가 쓰여 있는 돌을 몇 개 가져왔는지를 의미한다.)

위 조건을 만족하는 순서쌍을 찾을 수가 없다면, $-1\ 0$ 을 공백으로 구분하여 출력한다.

예제

표준 입력	표준 출력
3 2 3 1000 1000007	1003 1
4 5 6 7 8 8000019	1000003 5
3 100 101 302 419	-1 0

문제 F. 승고한 아이스크림 (icecream)

시간 제한: 2초
메모리 제한: 128 MB

Amel은 지구 온난화로 인하여 올해 여름이 무한히 길어진다는 소식을 들었다. Amel은 무한히 긴 여름을 버티기 위하여 새로 출시된 승고한 아이스크림을 대량 구매하기로 결심했다. 기쁜 마음으로 N 개의 서로 다른 승고한 아이스크림 박스를 샀지만 몇 가지 문제점이 있었다.

먼저 너무 짧은 기간에 아이스크림을 많이 먹으면 배탈이 난다. 아이스크림 박스마다 차가운 정도 C 에 따라 그 맛을 먹은 지 C 일째 되는 날 전까지는 어떤 아이스크림도 먹을 수 없다. 만일 이를 어기면 Amel은 배탈이 나고 만다. (C 는 0보다 크기 때문에 하루에는 한 가지 맛밖에 먹지 못한다.)

또한, 아이스크림은 시간이 지날수록 맛이 떨어진다. 아이스크림을 산 날 박스마다 맛있는 정도 D 가 주어진다. 아이스크림은 오래 놔두면 수분이 날아가 건조해지기 때문에, 하루가 지날 때마다 맛이 F 만큼 감소한다.

Amel은 배탈이 나지 않기 위해, 하루에 아이스크림을 한 스푼씩만 먹기로 결심했다. 하지만 아이스크림을 총 M 스푼보다 많이 먹게 된다면, 너무 질려버려 더 이상 먹지 못한다. 모든 박스는 적어도 M 스푼의 양은 있다. Amel은 먹은 아이스크림 맛의 합을 최대로 하고 싶다. Amel을 도와 승고한 아이스크림을 가장 맛있게 먹는 방법을 찾아주자!

입력 형식

첫 번째 줄에 승고한 아이스크림 박스의 종류 N , 먹을 수 있는 스푼 M , 하루마다 맛의 감소량 F 가 공백으로 구분되어 주어진다. ($0 < N \leq 100,000$, $0 < M \leq 5,000,000$, $0 < F \leq 10^9$)

두 번째 줄부터 N 개의 줄에 걸쳐, 각 박스의 C_i, D_i 가 공백으로 구분되어 주어진다. ($0 < C_i \leq 5,000,000$, $0 < D_i \leq 10^9$)

출력 형식

Amel이 먹을 수 있는 승고한 아이스크림 맛의 합의 최댓값을 출력한다.

예제

표준 입력	표준 출력
2 2 3 3 11 2 9	14
3 4 2 4 15 2 11 1 6	25

참고 사항

첫 번째 예제의 경우, 박스를 각각 A, B라 하고 먹는 경우에 따라서 맛의 합을 계산해보자.

(1일차 A, 4일차 A), (1일차 A, 4일차 B), (1일차 B, 3일차 A), (1일차 B, 3일차 B), (1일차 B, 4일차 A), (1일차 B, 4일차 B)

각각 맛의 합을 계산하면 $11 + 2, 11 + 0, 9 + 5, 9 + 3, 9 + 2, 9 + 0$ 로 최댓값은 14이다. 위에 제시된 방법이 모든 방법은 아니지만, 더 최적의 방법은 없다.

문제 G. 수열 복원 (restore)

시간 제한: 3초
메모리 제한: 256 MB

정수로 이루어진 길이 N 짜리 수열 a_1, a_2, \dots, a_N 과 덧셈으로 놀던 옥제는, 수열이 적혀있는 종이를 잃어버렸다. 다행히 가지고 놀던 M 개의 기록은 남아있다. 각각의 기록은 세 정수로 이루어져 있으며, 그 의미는 두 수의 인덱스 i, j 와 두 수의 합 $a_i + a_j$ 이다. 이 정보들을 토대로 원래의 수열이 무엇이었는지 복원하려고 한다.

입력 형식

첫 번째 줄에 수열의 길이 N 과 기록의 개수 M 이 주어진다. 두 번째 줄부터 M 개의 줄에 걸쳐 각각 세 정수 i, j, k 가 공백으로 구분되어 주어진다. 이는 $a_i + a_j = k$ 임을 의미한다. 답이 존재하지 않는 입력은 없다.
($1 \leq N, M \leq 500,000$, $1 \leq i, j \leq N$, $-10^9 \leq k \leq 10^9$)

출력 형식

옥제가 가지고 놀던 수열의 원소 N 개를 공백으로 구분하여 출력한다. 수열의 원소는 정수임이 보장된다. 답이 여러개 존재할 수 있다면 “ERROR”(따옴표 제외)를 출력한다.

예제

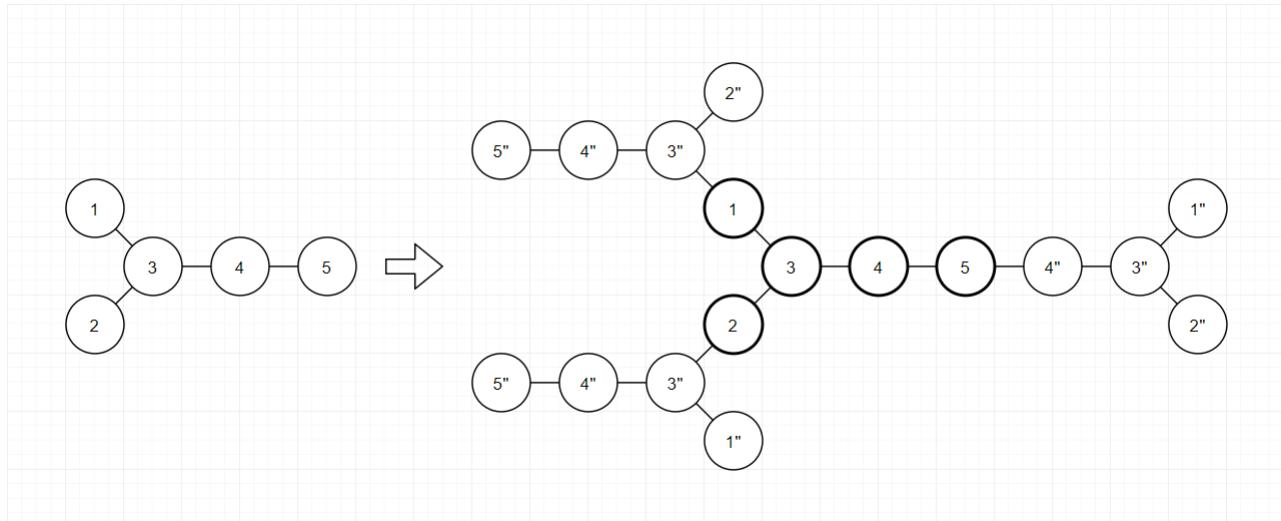
표준 입력	표준 출력
3 3 1 2 3 2 3 5 1 3 4	1 2 3
4 4 1 2 3 2 3 5 3 4 7 4 1 5	ERROR

문제 H. 반사복제된 트리 (mirrortree)

시간 제한: 1초
메모리 제한: 256 MB

어떤 트리 T 의 “반사복제” 행위를 다음과 같이 정의한다.

- T 의 각 리프 노드에 대하여, 그 리프 노드에다가 원래 트리 T 를 해당 리프 노드를 기준으로 동일한 위상으로 복제시킨다. 동일한 위상으로 복제시킨다는 것은, 어떤 리프 노드를 L 이라고 하면 트리를 통째로 복제한 후 두 트리의 L 에 해당하는 부분을 겹친다는 것을 의미한다.



이 그림은 주어진 트리(화살표 왼쪽)를 반사복제시켜서 만든 새로운 트리(화살표 오른쪽)이다. 굵은 원이 기존의 트리이며 얇은 원이 각 리프노드별로 동일한 위상으로 복제된 트리이다. 새로 생성된 노드는 기존의 노드와 구분하기 위해 번호에 쌍따옴표를 붙였다.

N 개의 노드를 가진 어떤 트리가 주어지고, 이 트리를 k 번 반사복제했을 때, 이 트리의 모든 노드 간 노드의 거리를 $10^9 + 7$ 로 나눈 나머지를 구하시오. 이때, 모든 노드 간 노드의 거리를 수식으로 표현하면 다음과 같다. (V 는 전체 노드의 집합, $dist$ 는 두 노드 사이의 거리(= 두 노드를 잇는 경로 상의 간선의 개수)를 의미한다.)

$$\frac{1}{2} \sum_{v_1 \in V} \sum_{v_2 \in V} dist(v_1, v_2)$$

입력 형식

첫 번째 줄에 트리에 있는 노드의 개수 N 과 이 트리를 반사복제할 횟수 K 가 공백으로 구분되어 주어진다. ($2 \leq N \leq 100,000$, $1 \leq K \leq 100,000$)

두 번째 줄부터 $N - 1$ 개의 줄에 걸쳐 두 정수 u_1, u_2 가 공백으로 구분되어 주어진다. 이는 트리의 u_1 번 노드와 u_2 번 노드가 서로 연결되어 있음을 의미한다. ($1 \leq u_1 < u_2 \leq N$) 간선들로 트리가 만들어지지 않는 입력은 주어지지 않는다.

출력 형식

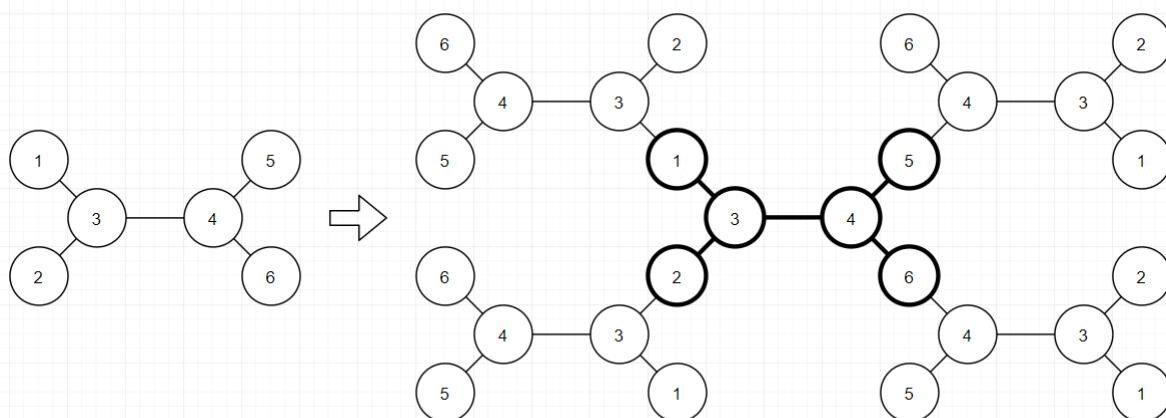
정답을 $10^9 + 7$ 로 나눈 나머지를 출력한다.

예제

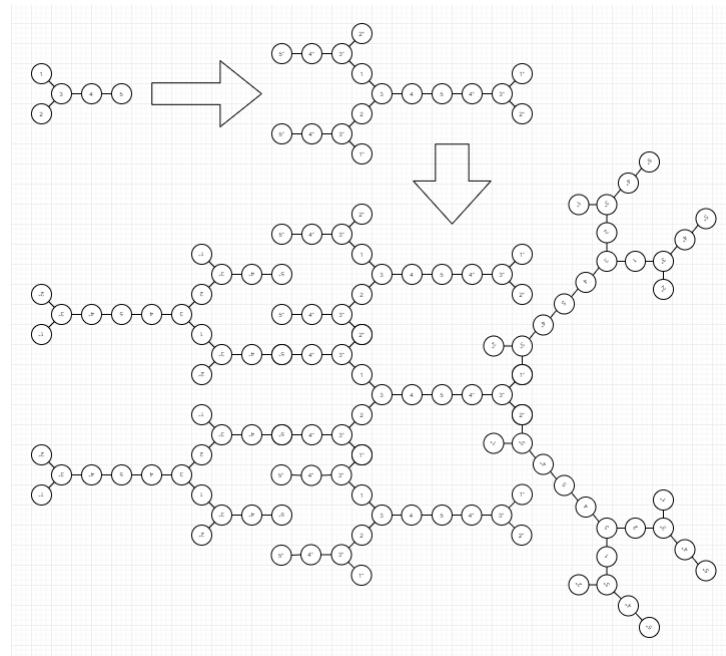
표준 입력	표준 출력
6 1 1 3 2 3 3 4 4 5 4 6	1645
5 2 1 3 2 3 3 4 4 5	89064

참고 사항

첫 번째 예제는 입력으로 주어진 트리를 1번 반사복제한 트리에 관하여 묻고 있다. 이를 그림으로 그리면 다음과 같다.



두 번째 예제는 입력으로 주어진 트리(= 본문에 있는 트리)를 2번 반사복제한 트리에 관하여 묻고 있다. 이를 그림으로 그리면 다음과 같다.



문제 I. 찾아라 드래곤볼! (dragonball)

시간 제한: 4.5 초
메모리 제한: 512 MB

2020년 4월 29일, 대한건아 권육제는 4주의 군사훈련을 마치고 수료를 했다!! 계속된 훈련으로 몸도 마음도 지친 육제는 집에 도착하자마자 편하게 낮잠을 자기 시작했다.

...

??? : 육제야...

...

태수 : 육제야... 일어나... 군대가야지...

육제 : ???

아뿔싸! 이 모든 게 꿈이었다니... 육제는 그 자리에 주저앉아 평평 울기 시작했다. 울고 있는 육제를 보아하니 마음이 아파진 태수는 육제에게 모종의 거래를 제안했다.

“이 마을에 있는 모든 드래곤볼을 모아오면 군대를 안 갈 수 있도록 힘 좀 써볼게!”

육제의 마을은 N 개의 정점과 $N - 1$ 개의 간선으로 이루어진 트리 형태를 하고 있고, 모든 간선의 길이는 1이다. 이 때 육제의 집은 1번 정점에 위치한다. 마을에 흩뿌려져 있는 모든 드래곤볼을 모아 다시 육제의 집으로 돌아오는데 필요한 최단거리를 구해주면 된다.

하지만 그냥 하면 육제가 쉽게 면제를 받을 것 같아서 기분이 나빠진 태수는 드래곤볼을 Q 번 추가하거나 없애기로 했다. 태수는 어떤 경로 상의 정점들에 드래곤볼 놓을 수도 있고, 경로 상의 정점들에 있는 모든 드래곤볼을 없앨 수 있다. 경로 상의 정점들에 드래곤볼을 추가하거나 없앨 때마다 육제의 집에서 출발하여 모든 드래곤볼을 모아 다시 육제의 집으로 돌아오는데 필요한 최단거리를 구해주면 된다.

처음에 육제의 마을엔 드래곤볼이 없다! 육제가 군대를 안가도 되도록 도와주자!

입력 형식

첫 번째 줄에 정점 개수 N 이 주어진다. ($2 \leq N \leq 100,000$)

두 번째 줄부터 $N - 1$ 개의 줄에 걸쳐 트리의 간선 정보를 의미하는 x, y 가 공백으로 구분되어 주어진다.

이는 x 와 y 가 연결되어있음을 의미한다. ($1 \leq x, y \leq N, x \neq y$)

주어진 입력은 트리를 구성한다.

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

$N + 2$ 번째 줄부터 Q 개의 줄에 걸쳐 쿼리에 대한 정보 k, u, v 가 공백으로 구분되어 주어진다. ($1 \leq k \leq 2, 1 \leq u, v \leq N$)

k 가 1일 때에는 u 정점과 v 정점을 잇는 경로상의 모든 정점에 드래곤볼을 1개 추가한다.

k 가 2일 때에는 u 정점과 v 정점을 잇는 경로상의 모든 정점에 있는 모든 드래곤볼을 없앤다.

출력 형식

Q 개의 줄에 걸쳐 쿼리에 대한 답을 출력한다. 모든 정점에 드래곤볼이 없다면 0을 출력한다.

예제

표준 입력	표준 출력
5	4
1 2	0
1 3	4
3 4	0
3 5	0
5	
1 1 4	
2 4 5	
1 2 3	
2 2 3	
1 1 1	