



전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2025

Preliminaries

Official Problemset

예선 문제

2025년 7월 12일, 14:00 – 17:00

주최

전국 대학생 프로그래밍 대회 동아리 연합

후원



나정휘 (jhnah917)

정재현(gravekper)

문제 목록

문제지에 있는 문제가 총 11문제가 맞는지 확인하시기 바랍니다.

- A** 체육은 수학과목 입니다 2
- B** 도미노 게임
- C** 빔
- D** 연산 추가하기
- E** 콘서트
- F** 시장조성하기
- G** 스마트 창고
- H** 해안선
- I** 로봇 청소기
- J** 소어그래프
- K** Distance Multiplication Maximization

모든 문제의 메모리 제한은 1GB로 동일합니다.

문제 A. 체육은 수학과목입니다 2

시간 제한 1 초 메모리 제한 1024 MB

UCPC 초등학교의 즐거운 체육 시간이 찾아왔다. 오늘은 운동장 5바퀴 달리기 수행평가가 있는 날이다.

달리기를 평소에 별로 좋아하지 않던 우경이는 출발 전부터 걱정이 태산이었다. 그런 우경이를 보고 체육 선생님은 따스한 미소로 말했다. "괜찮아, 너무 부담 갖지 마. 천천히 뛰어도 괜찮으니까, 5바퀴를 1800초 이하의 시간에 완주하면 돼."

선생님의 다정한 응원 덕분에 용기를 얻은 우경이는 천천히 힘을 내어 달리기 시작했다. 조금 힘들었지만, 포기하지 않고 한 바퀴씩 꾸준히 달려 어느덧 4바퀴를 완주했다.

이제 마지막 1바퀴. 땀을 닦으며 잠시 숨을 고른 우경이는 문득 걱정이 들었다. "내가 너무 천천히 뛰었나...? 마지막 한 바퀴만 잘 뛰면 수행평가를 통과할 수 있을까?" 지친 상태라 마지막 바퀴는 분명 300초 이상 걸릴 것 같았지만, 우경이는 최선을 다해 딱 300초에 맞춰 달리기로 마음먹었다.

지금까지 우경이가 뛴 4바퀴의 시간이 초 단위로 각각 주어질 때, 마지막 한 바퀴를 정확히 300초에 뛴다면 수행평가를 무사히 통과할 수 있을지 계산해보자.

입력

4개의 줄에 걸쳐 우경이가 각 바퀴를 뛰는 데 걸린 시간 t_i 가 초 단위로 한 줄에 하나씩 주어진다. ($1 \leq t_i \leq 1800$)

출력

우경이가 마지막 바퀴를 뛰어서 수행평가를 무사히 통과할 수 있으면 Yes, 불가능하면 No를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
375 375 375 375	Yes
375 375 375 400	No
300 300 300 300	Yes

문제 B. 도미노 게임

시간 제한 1 초 메모리 제한 1024 MB

$N \times M$ 크기의 격자가 주어진다. i 행 j 열의 칸에는 음이 아닌 정수 a_{ij} 가 적혀 있다. ($1 \leq i \leq N; 1 \leq j \leq M$)
민지는 각 차례마다 아래와 같은 행동을 한다.

1. 격자에서 상하좌우로 인접한 두 칸을 선택한다. 이때 두 칸에 적힌 수 중 적어도 하나는 양의 정수이어야 한다.
2. 선택한 칸에 적힌 수에서 1씩 뺀다. 만약 어떤 칸에 적힌 수가 0이라면 그대로 둔다.

모든 칸에 적힌 수가 0이 될 때 게임이 종료된다. 민지는 이 게임을 최대한 오래 하려고 한다. 민지가 최선을 다해 게임을 오래 진행했을 때, 진행할 수 있는 최대 차례의 수를 구해보자.

입력

첫째 줄에 양의 정수 N, M 이 공백으로 구분되어 주어진다. ($2 \leq N, M \leq 1000$)

이후 N 개의 줄에 걸쳐 격자에 적힌 수가 주어진다. 그 중 i 번째 줄에는 $a_{i1}, a_{i2}, \dots, a_{iM}$ 이 공백으로 구분되어 주어진다. ($0 \leq a_{ij} \leq 10^9$)

출력

진행할 수 있는 최대 차례의 수를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2 2 0 1 0 0	1
2 2 1 0 1 1	3

문제 C. 빔

시간 제한 2 초 메모리 제한 1024 MB

당신은 구간 보관 서비스를 운영하고 있다. 현재 보관소에는 총 N 개의 구간이 보관되어 있고, 이 중 i 번째 구간은 수직선에서 $[l_i, r_i]$ 로 나타난다.

보관소에 총 Q 회의 레이저 폭격이 예고되었다! 이 중 j 번째 레이저 빔의 피해 범위는 구간 $[s_j, e_j]$ 로 나타낼 수 있다. 당신은 각 레이저 빔이 날아올 때마다, 보관된 구간을 적절히 옮겨서 빔에 맞는 구간이 없도록 해야 한다.

구체적으로, 보관된 각 구간 $[l_i, r_i]$ 에 대해 적절한 정수 x_{ij} 를 정한다. 이 때 모든 (i, j) 에 대해 $[l_i + x_{ij}, r_i + x_{ij}]$ 와 $[s_j, e_j]$ 가 겹치는 부분의 길이가 0이 되도록 해야 한다. 두 구간 $[a, b]$ 와 $[c, d]$ 가 겹치는 부분의 길이는 $\max(0, \min(b, d) - \max(a, c))$ 이다.

구간은 무겁기 때문에 기계를 사용하여 옮기는데, 한 번 옮길 때마다 (이동 거리) \times (구간 길이) 만큼의 전기료가 나온다. 즉, j 번 레이저 빔이 오기 전에 사용하는 전기료는 $\sum_{i=1}^N (r_i - l_i) |x_{ij}|$ 이다.

각 레이저 빔이 지나간 후에 당신은 옮겼던 모든 구간을 다시 원래 위치로 되돌려 놓는다. 옮길 때와 돌려놓을 때 모두 똑같이 비용이 발생함을 유의하라.

당신은 각 레이저 빔마다 최소한의 전기료를 사용하여 모든 구간을 안전하게 관리하려고 한다. 이때의 비용을 계산하여 보자.

입력

첫째 줄에 구간의 개수 N 과 예고된 레이저 폭격의 수 Q 가 주어진다. ($1 \leq N, Q \leq 250000$)

이후 N 개의 줄에 걸쳐, 그 중 i 번째 줄에는 i 번째 보관된 구간의 양 끝점 l_i 와 r_i 가 주어진다. ($1 \leq l_i < r_i \leq 1000000$)

이후 Q 개의 줄에 걸쳐, 그 중 j 번째 줄에는 j 번째 레이저 빔 피해 범위의 양 끝점 s_j 와 e_j 가 주어진다. ($1 \leq s_j < e_j \leq 1000000$)

입력으로 들어오는 모든 수는 정수이다.

출력

각 레이저 빔에 대해, 현재 상태에서 모든 구간을 안전한 위치로 옮겼다가 돌려놓기 위해 필요한 최소 전기료를 Q 개의 줄에 순서대로 출력한다.

입출력 예시

표준 입력(stdin)		표준 출력(stdout)
2	2	24
1	5	0
4	8	
3	5	
8	9	

문제 D. 연산 추가하기

시간 제한 1 초 메모리 제한 1024 MB

FuriosaAI는 데이터센터 및 고성능 엣지를 위한 AI 반도체를 개발한다. 2세대 제품 RNGD는 자체 설계한 아키텍처인 TCP(Tensor Contraction Processor) 기반의 인공지능 가속기로, PyTorch 등 주요 프레임워크와 호환되는 SDK를 제공한다.

RNGD에서 인공지능 모델의 추론을 수행하려면, 먼저 모델에서 수행하는 계산을 RNGD 명령어들의 나열로 변환하는, 즉 컴파일하는 과정이 필요하다. 실제 컴파일 과정을 단순화한 다음과 같은 상황을 가정해보자.

RNGD 연산 시간의 기본 단위는 사이클로, 모델의 실행은 총 H 사이클 이내에 끝나야 한다는 제한이 있다. H 개의 사이클을 순서대로 사이클 1, 사이클 2, ..., 사이클 H 라고 부르자.

컴파일된 인공지능 모델에는 총 N 개의 연산이 있으며, 이 중 i 번째 연산은 사이클 A_i 부터 사이클 B_i 까지 실행된다. 실행되는 사이클이 서로 겹치는 연산들이 있을 수도 있다.

RNGD에서 컴파일된 모델을 실행하는 도중에 추가 연산 하나를 함께 수행해야 한다는 요청이 들어왔다. 이때 추가 연산은 기존 N 개의 연산 중 어느 것과도 실행 사이클이 겹치면 안 되며, H 사이클 이내에서 수행이 완료되어야 한다. 구체적으로,

- 추가 연산을 수행하는 데 T 사이클이 걸린다고 하자. 즉, 추가 연산이 사이클 S 에 시작된다면, 사이클 $S + T - 1$ 에 종료된다.
- 추가 연산이 수행되는 T 개의 사이클 동안, 다른 연산이 동시에 수행되어서는 안 된다.
- $1 \leq S \leq S + T - 1 \leq H$ 여야 한다.

추가 연산의 종류에 따라 소요되는 사이클 수 T 가 달라질 수 있으므로, Q 가지 시나리오를 미리 검토하려 한다. i 번째 시나리오에서 추가 연산을 수행하는 데 T_i 사이클이 필요한 경우, 조건을 만족하는 시작 사이클 S 가 총 몇 개인지 빠르게 계산해보자.

각 시나리오의 추가 연산은 누적되지 않음에 유의하라.

입력

첫째 줄에 인공지능 모델을 구성하는 연산의 수 N , 전체 사이클 수 H 가 공백으로 구분되어 주어진다. ($0 \leq N \leq 200\,000$; $1 \leq H \leq 10^9$)

이후 N 개의 줄에 걸쳐, 그 중 i 번째 줄에 i 번째 연산이 실행되는 시작 사이클과 종료 사이클을 가리키는 두 정수 A_i, B_i 가 공백으로 구분되어 주어진다. ($1 \leq A_i \leq B_i \leq H$)

그 다음 줄에 시나리오의 개수 Q 가 주어진다. ($1 \leq Q \leq 200\,000$)

이후 Q 개의 줄에 걸쳐, 추가 연산의 소요 사이클 수 T_i 가 주어진다. ($1 \leq T_i \leq H$)

출력

Q 개의 줄에 각 시나리오에 대한 정답을 순서대로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 30	11
5 8	5
12 19	2
14 23	
3	
2	
4	
6	
0 4	4
2	3
1	
2	

문제 E. 콘서트

시간 제한 1 초 메모리 제한 1024 MB

현빈이가 사는 도시에는 날마다 콘서트가 열린다.

현빈이는 소음에 민감해 N 개의 방음벽을 설치하였다. 방음벽은 순서대로 1번부터 N 번까지의 번호를 가지며, i 번 방음벽은 D_i 만큼의 소음을 흡수할 수 있다. 방음벽을 설치한 뒤, 현빈이는 앞으로 모든 콘서트는 두 연속한 방음벽 사이에서만 열 수 있도록 규칙을 세웠다. 즉, 앞으로 모든 콘서트가 열리는 위치는 어떤 정수 c 에 대해 c 번 방음벽과 $c+1$ 번 방음벽 사이여야 한다. ($1 \leq c < N$)

모든 소음을 흡수해 조용한 나날을 보내려던 현빈이의 계획과 달리, 방음벽이 콘서트의 소음을 완전히 감당하지 못하는 일이 생기기도 했다. 예를 들어, x 의 소음을 가진 콘서트가 c 와 $c+1$ 번 방음벽 사이에서 열렸다고 하자. 이때 c 번 방음벽이 흡수할 수 있는 소음은 $\min(D_c, x)$ 뿐이고, 만약 흡수되지 못한 소음이 있다면 해당 소음은 $c-1$ 번 방음벽으로 향하게 된다. 마찬가지로, $c+1$ 번 방음벽 또한 $\min(D_{c+1}, x)$ 만큼의 소음만 흡수하고, 흡수되지 못한 소음은 $c+2$ 번 방음벽으로 향하게 된다. 이 과정은 더 이상 흡수될 소음이 없거나 소음을 흡수할 방음벽이 남아 있지 않을 때까지 반복된다.

현빈이는 매 콘서트가 끝난 직후, N 개의 모든 방음벽에 대하여 각 방음벽이 흡수한 소음의 양만큼 방음벽을 보강할 것이다. 즉, 어떤 방음벽 k 가 흡수한 소음이 x 라면, 콘서트가 끝난 직후 k 번 방음벽이 흡수할 수 있는 소음은 $D_k + x$ 가 된다.

현빈이는 Q 회에 걸쳐 아래 작업 중 하나를 진행한다.

- 1 c x : c 와 $c+1$ 번 방음벽 사이에서 소음 x 의 콘서트가 열려, 방음벽을 보강한다. ($1 \leq c < N$; $1 \leq x \leq 10^9$)
- 2 c : c 번 방음벽이 흡수할 수 있는 소음의 크기를 측정한다. ($1 \leq c \leq N$)

2번 작업을 진행할 때마다 주어진 방음벽이 흡수할 수 있는 소음을 빠르게 계산해 보자.

입력

첫째 줄에 현빈이가 설치한 방음벽의 수 N 이 주어진다. ($2 \leq N \leq 200000$)

둘째 줄에 N 개의 방음벽이 흡수할 수 있는 소음의 크기 D_1, D_2, \dots, D_N 이 공백으로 구분되어 주어진다. ($1 \leq D_i \leq 10^9$)

셋째 줄에 현빈이가 진행한 작업의 수 Q 가 주어진다. ($1 \leq Q \leq 200000$)

이후 Q 개의 줄에 걸쳐, 현빈이가 진행할 작업에 대한 정보가 지문과 같은 형식으로 주어진다. 2번 작업이 최소 한 번 이상 주어짐이 보장된다.

입력으로 들어오는 모든 수는 정수이다.

출력

Q 개의 줄에 현빈이가 진행한 모든 2번 작업의 결과를 순서대로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6	3
5 1 2 4 7 3	6
5	14
1 2 1	
2 3	
1 4 7	
2 3	
2 5	

문제 F. 시장조성하기

시간 제한 4 초 메모리 제한 1024 MB

Hudson River Trading(HRT)은 뛰어난 수학, 기술적 역량을 활용하여 거래 전략을 설계하는 퀀트 트레이딩 회사이다. HRT는 엔지니어와 연구자들이 한 팀을 이뤄 어려운 문제를 해결하며, 글로벌 금융 시장에서 매일 수백만 주의 주식을 거래한다.

HRT의 신입 알고리즘 개발자는 새로 설계한 거래 전략을 따르는 시장조성(market-making) 엔진을 개발한 뒤, 그 엔진의 안정성을 평가하고자 한다. 시장조성 엔진은 주식을 보유하지 않은 가상의 계좌를 이용하여 거래를 시작하며, N 개의 연속된 매매 시점(틱)에 아래와 같은 전략을 따라 거래한다.

- i 번째 매매 시점에 엔진은 a_i 이상 b_i 이하의 정수 하나를 선택한다. 선택한 정수가 양수라면 그만큼 주식을 매수하고, 음수라면 선택한 정수의 절댓값만큼 주식을 매도하고, 0이라면 주식을 매매하지 않는다. 이 때 보유한 주식이 부족해도 매도가 가능하며, 이 경우 보유 주식의 수는 음수가 된다.
- i 번째 매매 직후 보유 주식이 0주라면, 해당 전략이 포지션 노출을 최소화하여 위험성을 완화하고 시장 안정성에 기여했다고 평가하여 시장조성 엔진의 안정성이 x_i 만큼 증가한다.
- 수수료 또는 슬리피지 등 다른 요소는 모두 무시한다.

총 N 번의 매매를 모두 마쳤을 때, 신규 개발한 시장조성 엔진이 달성할 수 있는 안정성의 최댓값을 구해보자.

입력

첫째 줄에는 주식을 매매한 횟수 N 이 주어진다. ($1 \leq N \leq 1000000$)

다음 N 개의 줄에 걸쳐, 그 중 i 번째 줄에 i 번째 매매에 대한 정보를 의미하는 세 정수 a_i, b_i, x_i 가 공백으로 구분되어 주어진다. ($-10^9 \leq a_i \leq b_i \leq 10^9$; $1 \leq x_i \leq 10^9$)

출력

총 N 번의 매매를 모두 마쳤을 때, 신규 개발한 시장조성 엔진이 달성할 수 있는 안정성의 최댓값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 -1 0 3 1 1 2 -1 0 5	8
5 1 1 1000 -2 -1 7 1 1 5 -1 -1 4 1 1 8	13
8 -1 1 5 -4 2 7 3 4 4 -6 4 8 -2 -1 6 -5 7 1 4 6 9 -7 7 5	34

문제 G. 스마트 창고

시간 제한 3 초 메모리 제한 1024 MB

모빌리티 혁신과 미래를 선도하는 기업, 현대오토에버에서는 4차 산업혁명 미래에 맞춰 스마트 물류 패러다임을 바꾸어나가고 있다.

현대오토에버가 새로 개발해 테스트 중인 스마트 창고에는 $N \times M$ 크기의 격자 형태로 화물이 배치되어 있다. 격자의 위에서부터 r 번째 행, 왼쪽에서부터 c 번째 열의 칸을 (r, c) 와 같이 표기할 때, 칸 (r, c) 에는 가치가 A_{rc} 인 화물이 있다. 화물의 가치는 음수일 수도 있다.

창고의 로봇은 가끔 특정 화물을 창고 밖으로 운반하는 지시를 받는다. 이때 효율적인 운반을 위해, 로봇은 해당 화물뿐만 아니라 근처에 있는 다른 화물들을 직사각형 형태로 한번에 운반할 수 있다. 구체적으로, 칸 (r, c) 에 있는 화물을 운반하라는 지시를 받았을 경우, 로봇은 $1 \leq r_1 \leq r \leq r_2 \leq N$, $1 \leq c_1 \leq c \leq c_2 \leq M$ 을 만족하는 네 정수 r_1, r_2, c_1, c_2 를 고른 뒤, 직사각형 영역 $[r_1, r_2] \times [c_1, c_2]$ 안에 있는 모든 화물들을 운반한다.

NM 개의 화물 각각에 대해, 로봇이 해당 화물을 운반하라는 지시를 받았을 때, 운반할 수 있는 화물의 가치 합의 최댓값을 구해 보자.

입력

첫째 줄에는 창고의 크기를 나타내는 두 정수 N 과 M 이 공백으로 구분되어 주어진다. ($2 \leq N \leq 500$; $2 \leq M \leq 500$)

이후 N 개의 줄에 걸쳐, 그 중 r 번째 줄에는 격자의 r 번째 행에 있는 각 화물의 가치를 나타내는 M 개의 정수 $A_{r1}, A_{r2}, \dots, A_{rM}$ 이 공백으로 구분되어 주어진다. ($-10^3 \leq A_{rc} \leq 10^3$)

출력

N 개의 줄에 걸쳐, r 번째 줄에 칸 $(r, 1), (r, 2), \dots, (r, M)$ 에 대해 각각의 화물을 운반하라는 지시를 받았을 때 로봇이 운반할 수 있는 화물의 가치 합의 최댓값을 의미하는 M 개의 정수를 공백으로 구분해 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 4	8 8 9 8
2 2 2 2	6 1 9 4
2 -8 0 -12	6 6 12 12
2 -8 7 5	
2 2	1 -3
-1 -3	2 1
2 -1	

문제 H. 해안선

시간 제한 1 초 메모리 제한 1024 MB

N 개의 도시가 해안선을 따라 원형으로 배치되어 있다. 도시는 시계 방향으로 1번부터 N 번까지 번호가 차례로 붙어 있다. 모든 도시 쌍에 대해, 해당 두 도시를 연결하는 양방향 직선 도로가 존재한다.

이 수많은 도로들 중, a 번 도시와 b 번 도시를 잇는 도로를 특별한 도로라고 부른다. 이 특별한 도로는 1번 도시가 아닌 두 도시를 이으며, 경치가 가장 아름다운 도로로 알려져 있다.

정서는 다음과 같은 방식으로, 도로만을 이용해 이 도시들을 여행하려 한다.

- 여행은 1번 도시에서 출발하며, 출발과 동시에 1번 도시는 이미 방문한 것으로 간주한다.
- 여행 중에는 1번 도시를 포함한 이미 방문한 도시를 다시 방문해서는 안 되며, 모든 도시를 정확히 한 번씩 방문해야 한다.
- 여행 중 이용하는 어떤 두 도로도 서로 교차하지 않아야 한다.
- 여행 중에는 반드시 특별한 도로를 한 번 이용해야 한다. 단, 특별한 도로를 이용하는 방향은 중요하지 않다.

위 조건을 모두 만족하는 이동 경로의 수를 구하시오.

입력

첫째 줄에 도시의 수 N 이 주어진다. ($3 \leq N \leq 1000000$)

둘째 줄에 특별한 도로가 있는 두 도시의 번호 a, b 가 공백으로 구분되어 주어진다. ($2 \leq a, b \leq N; a \neq b$)

출력

조건을 만족하는 모든 경로의 수를 1000000007로 나눈 나머지를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 2 4	2
6 5 4	11

문제 I. 로봇 청소기

시간 제한 1 초 메모리 제한 1024 MB

LG ThinQ는 LG전자의 AI 플랫폼으로, 가전제품의 상태를 실시간으로 모니터링하고 사용자에게 맞춤형 서비스를 제공한다.

LG ThinQ AI를 탑재한 로봇 청소기가 정사각형 모양의 방안에 놓여 있다. 이 로봇은 방바닥을 $10^9 \times 10^9$ 크기의 격자로 나누어 각 칸에 오염물질이 있는지를 판단할 수 있다. 편의상 오염물질이 있는 칸을 오염된 칸이라고 하자. 위에서 y 번째 가로줄에서 왼쪽에서 x 번째에 있는 칸의 좌표를 (x, y) 라고 할 때, 이 로봇은 N 개의 서로 다른 오염된 칸의 위치 정보를 다음의 압축 과정을 거쳐 서버로 전송한다.

- 오염된 칸들의 좌표 (x, y) 를 x 좌표가 증가하는 순서로, x 좌표가 같으면 y 좌표가 증가하는 순서로 정렬한다.
- 이렇게 정렬된 좌표들의 x 좌표를 제거하고 y 좌표만 순서대로 서버로 보낸다.

두 오염된 칸이 서로 상하좌우로 인접하면 같은 오염 영역에 속한다고 할 때, 서버는 로봇이 보낸 데이터를 받아서 오염 영역의 개수를 추측하여야 한다. 서버가 받은 데이터가 주어질 때, 가능한 오염 영역의 개수의 최솟값과 최댓값을 구하시오.

입력

첫째 줄에는 오염된 칸의 개수 N 이 주어진다. ($1 \leq N \leq 200\,000$)

다음 N 줄에 걸쳐, 서버가 받은 오염된 칸들의 y 좌표를 의미하는 N 개의 정수 y_1, \dots, y_N 이 한 줄에 하나씩 순서대로 주어진다. ($1 \leq y_i \leq 10^9$)

출력

첫째 줄에 오염 영역의 개수의 최솟값을 출력한다.

둘째 줄에 오염 영역의 개수의 최댓값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6	1
1	6
3	
4	
1	
2	
3	

문제 J. 소어그래프

시간 제한 1 초 메모리 제한 1024 MB

정점의 개수가 N 인 소어그래프(XOR Graph)는 다음과 같이 정의된다.

- 소어그래프는 0부터 $N-1$ 까지의 번호가 붙은 N 개의 정점을 가진다.
- 각 정점 i 에 대해, 정점 i 에서 정점 $i \oplus t$ 와 정점 $(i \oplus t) + 1$ 로 가는 방향 간선이 존재한다.
- 단, 간선을 통해 이동하려는 도착 정점의 번호가 $N-1$ 을 초과하는 경우, 해당 간선은 존재하지 않는다.

여기서 \oplus 기호는 Bitwise XOR을 나타낸다.

정점의 개수 N , 시작 정점 x , 도착 정점 y , 그리고 음이 아닌 정수 t 가 주어질 때, 정점 x 에서 정점 y 로 이동하기 위해 지나야 하는 간선의 최소 개수를 구하는 프로그램을 작성하라.

입력

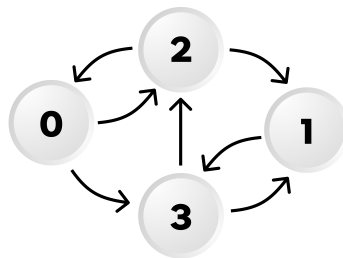
첫째 줄에 정수 N, x, y, t 가 공백으로 구분되어 주어진다. ($2 \leq N \leq 10^{18}$; $0 \leq x, y < N$; $x \neq y$; $0 \leq t < 2^{20}$)

출력

정점 x 에서 정점 y 로 이동하기 위해 지나야 하는 간선의 최소 개수를 출력한다. 만약 정점 x 에서 정점 y 로 가는 경로가 존재하지 않는다면 -1 을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 2 3 2	2



표준 입력(stdin)	표준 출력(stdout)
10 7 9 6	-1
165321961421 998244353 5029393147 98207	8242633

문제 K. Distance Multiplication Maximization

시간 제한 7 초 메모리 제한 1024 MB

N 개의 정점으로 이루어진 트리(사이클이 없는 무방향 연결 그래프)가 있다. 정점은 1번부터 N 번까지 번호가 매겨져 있고, 간선은 1번부터 $(N-1)$ 번까지 번호가 매겨져 있다.

아래의 쿼리를 수행하는 프로그램을 작성하시오.

- $u\ v$: 정점 $x(1 \leq x \leq N)$ 에 대해, $\text{dist}(x, u) \times \text{dist}(x, v)$ 의 최댓값을 출력한다. ($1 \leq u, v \leq N$)

이때 $\text{dist}(x, y)$ 는 정점 x 에서 정점 y 로 가는 최단 경로 상의 간선 개수로 정의한다. 트리의 모든 정점 x 에 대해 $\text{dist}(x, x) = 0$ 이다.

입력

첫째 줄에 트리의 정점 수 N 가 주어진다. ($2 \leq N \leq 300\,000$)

다음 $(N-1)$ 개의 줄에는 트리의 정보가 주어진다. 이중 i 번째 줄에는 i 번 간선이 연결하는 두 정점 번호가 공백을 사이에 두고 주어진다.

다음 줄에 쿼리의 수 Q 가 주어진다. ($2 \leq Q \leq 300\,000$)

다음 줄부터 Q 개의 줄에는 쿼리의 정보가 한 줄에 하나씩 주어진다.

출력

Q 개의 줄에 쿼리의 답을 순서대로 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5	6
1 2	3
2 3	9
2 4	
4 5	
3	
1 2	
2 5	
3 3	