



UCPC 2023 본선

Official Problemset

전국 대학생 프로그래밍 대회 동아리 연합
여름대회 2023

주최 전국 대학생 프로그래밍 대회 동아리 연합

후원  **STARTLINK**



HYUNDAI
MOBIS

 **MOLOCO**

HYUNDAI
AutoEver

 **DEVOCEAN**

SOLVED. 

 **NEXON**

 **Jane Street®**

문제 목록

문제지에 있는 문제가 총 13문제가 맞는지 확인하시기 바랍니다.

- A** 계통수 추론
- B** I forgot 🧠
- C** 황혼
- D** 거대 로봇 전투
- E** 격자 속의 직선 경로
- F** 두 트리
- G** K번째 스페이드 찾기
- H** 피보나치 반반수열
- I** 안테나 설치
- J** 회전초밥
- K** 그건 가지가 아니라 대파예요
- L** 나무늘보
- M** 산 색칠

모든 문제의 메모리 제한은 1024 MB로 동일합니다.

문제 A. 계통수 추론

시간 제한 2 초 메모리 제한 1024 MB

계통수(phylogenetic tree)는 다양한 종 또는 개체의 유사성, 물리적 특성, 유전적 특성 등의 차이에 근거하여 유추된 진화적 관계를 보여주는 다이어그램이며, 루트 정점이 있는 트리로 표현된다.

지연이는 오랜 세월의 연구를 통해 N 개의 종 사이 상관관계를 규명하는 가설 M 개를 제안해 냈다. 각 종은 1부터 N 까지의 서로 다른 양의 정수로 구분된다. 지연이는 연구를 검증하기 위해 i 번 종이 정점 i 에 대응되는 계통수를 만들어보고자 한다.

루트 정점이 있는 트리의 두 정점 x 와 y 에 대해, 두 정점이 같거나 정점 x 가 정점 y 의 부모 정점의 조상일 때 정점 x 는 정점 y 의 조상이다. 정점 x 가 정점 y 의 조상이면 정점 y 는 정점 x 의 후손이며 그 역도 성립한다. 두 정점의 최소공통조상은 두 정점을 후손으로 가지는 정점 중 조상의 수가 가장 많은 정점이다.

지연이가 연구를 통해 제안한 가설은 각각 (a, b, c, d) 꼴로 주어진다. 각 가설은 계통수의 정점 a 와 정점 b 의 최소공통조상이 정점 x 이고 정점 c 와 정점 d 의 최소공통조상이 정점 y 일 때, 정점 x 가 정점 y 의 후손이며 정점 x 와 정점 y 가 다르다고 주장한다.

N 개의 종에 대한 M 개의 가설이 주어질 때 i 번 종이 정점 i 에 대응되는 계통수를 만들어보자. 모든 조건을 충족하는 계통수가 존재한다면 모든 정점의 번호가 1 이상 $2N$ 이하인 계통수가 존재함을 보일 수 있다.

입력

첫 번째 줄에 지연이가 연구한 종의 개수 N 과 가설의 개수 M 이 공백으로 구분되어 주어진다. ($4 \leq N \leq 2000$; $1 \leq M \leq 2000$)

이후 M 개의 줄에 걸쳐 각 가설을 의미하는 네 정수 a, b, c, d 가 공백으로 구분되어 주어진다. ($1 \leq a, b, c, d \leq N$; $a \neq b$; $c \neq d$)

출력

모든 가설을 충족하는 계통수 T 가 존재하지 않으면 첫째 줄에 **-1**을 출력한다. 그렇지 않으면, 첫째 줄에 계통수 T 를 구성하는 정점의 개수 N' ($N \leq N' \leq 2N$)을 출력한다. 이때 계통수 T 는 정점 1, 정점 2, ..., 정점 N' 으로 구성된다. 둘째 줄에 N' 개의 정수 $p_1, p_2, \dots, p_{N'}$ 을 공백으로 구분하여 출력한다. 정점 i 가 T 의 루트 정점일 때 p_i 는 0이며, 그 외에는 정점 i 의 부모 정점의 번호이다.

가능한 계통수가 여러 가지일 경우 아무거나 하나 출력한다.

입출력 예시

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

문제 B. I forgot

시간 제한 3 초 메모리 제한 1024 MB

무메이는 기억력 향상을 위해 매일매일 같은 그림 찾기 게임으로 수련하고 있다. 이 게임에는 1부터 N 까지의 정수가 적힌 카드가 수마다 두 장씩 있다. 이 카드들을 섞고 책상에 일렬로 내려놓아 수가 보이지 않게 한다. 턴마다 무메이는 카드를 두 장 골라 뒤집은 다음, 두 카드에 쓰인 수가 같다면 둘을 책상에서 치우고, 아니라면 도로 뒤집어 놓는다. 이를 반복해서 모든 카드를 책상에서 치우는 것이 목표이다.

때는 2043년. 무메이는 이 게임에서 어떤 실수도 범하지 않을 정도로 누구보다 뛰어난 기억력을 얻게 되었다. 이제 무메이는 다음과 같이 정해진 절차로 게임을 진행한다.

- 같은 수가 쓰인 카드 한 쌍을 알고 있다면, 그 두 카드를 뒤집고 책상에서 치운다.
- 그렇지 않다면, 우선 뒤집어 본 적 없는 카드 중 맨 왼쪽에 있는 것을 뒤집는다.
 - 해당 카드와 같은 수가 쓰인 다른 카드를 알고 있다면, 그 카드를 뒤집고 둘을 책상에서 치운다.
 - 그렇지 않다면, 여전히 지금까지 뒤집어 본 적 없는 카드 중 맨 왼쪽에 있는 것을 뒤집는다. 두 카드에 적힌 수가 같다면 둘을 책상에서 치우고, 아니라면 도로 뒤집어 놓는다.

위 절차를 따르면 턴마다 고를 카드 쌍이 유일하게 결정됨을 증명할 수 있다.

당신은 책상의 카드를 탐지해 게임을 마치는 데 필요한 턴 수를 미리 알려주는 프로그램을 작성하기로 했다. 그런데 게임이 너무 재미없다고 생각한 장난꾸러기 벨즈가 카드 쌍을 아무렇게나 서로 바꾸기 시작했다. 카드가 바뀔 때마다 필요한 턴 수를 구하는 프로그램을 작성하여라.

입력

첫 줄에 카드의 종류 수 N 과 카드를 바꾸는 횟수 Q 가 공백으로 구분되어 주어진다. ($1 \leq N, Q \leq 200000$)

다음 줄에 일렬로 놓인 카드에 적힌 수가 공백으로 구분되어 차례대로 주어진다. 모든 수는 1 이상 N 이하의 정수이며, 각 정수는 정확히 두 번씩 등장한다.

다음 Q 줄에 l 번째와 r 번째 카드를 서로 바꿨음을 나타내는 정수 l 과 r 이 공백으로 구분되어 주어진다. 모든 행동의 영향은 누적된다. 즉 모든 i 에 대해 초기 상태에서 $1, 2, \dots, i-1$ 번째 행동이 차례대로 시행된 다음에 i 번째 행동이 시행된다. ($1 \leq l < r \leq 2N$)

출력

첫 줄에는 초기 상태에서 게임을 마치는 데 필요한 턴 수를 출력한다.

다음 줄부터 Q 줄에 걸쳐 각 행동이 시행된 뒤에 게임을 마치는 데 필요한 턴 수를 출력한다.

입출력 예시

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

문제 C. 황혼

시간 제한 4 초 메모리 제한 1024 MB

오스타니아는 N 개의 도시와 이를 잇는 M 개의 도로로 구성된 나라이다. 도시는 1번부터 N 번까지 차례대로 번호가 붙어 있으며, i ($1 \leq i \leq M$)번째 도로는 u_i 번 도시에서 v_i 번 도시로 가는 단방향 도로이다. i 번째 도로를 거쳐서 이동하면 w_i 만큼의 시간이 소요된다. 임의의 서로 다른 두 도시에 대해 한 도시에서 다른 도시로 가는 도로는 1개 이하이다. 편의상 u 번 도시에서 v 번 도시로 가는 도로를 거쳐서 이동하는 데 걸리는 시간을 $w(u, v)$ 라고 하자.

웨스탈리스는 오스타니아와 냉전 관계에 놓여있는 국가이다. 웨스탈리스는 여러 첩보원을 통해 오스타니아에 대한 K 개의 정보를 얻었다. 각각의 정보는 하나의 단순 경로로 표현된다. 엄밀히 말해, j ($1 \leq j \leq K$)번째 정보는 s_j 개의 서로 다른 도시의 나열 $(p_{j,1}, p_{j,2}, \dots, p_{j,s_j})$ 로 구성되며, $p_{j,t}$ 번 도시에서 $p_{j,t+1}$ 번 도시로 가는 도로가 존재한다. ($1 \leq t < s_j$) 어떤 도시도 서로 다른 두 개의 정보에 동시에 속해 있지 않다.

웨스탈리스의 첩보원 코드네임 <황혼>은 현재 오스타니아의 1번 도시에 있다. 웨스탈리스 정부는 정보 수집을 위해 황혼에게 x 번 도시까지 이동하라는 지시를 내렸다. 웨스탈리스 최고의 스파이인 황혼은 정보 수집의 효율을 최대화하기 위해 K 개의 단순 경로 중 어느 것도 포함하지 않는 경로로 이동하기로 했다. 구체적으로, 황혼이 방문한 도시를 차례대로 (q_1, q_2, \dots, q_l) 이라고 하자. 이때, 다음 조건을 모두 충족해야 한다.

1. q_i 번 도시에서 q_{i+1} 번 도시로 가는 도로가 존재한다. ($1 \leq i < l$)
2. $(p_{z,1}, p_{z,2}, \dots, p_{z,s_z}) = (q_i, q_{i+1}, \dots, q_{i+s_z-1})$ 를 충족하는 정수 i 와 z 가 존재하지 않는다.
3. 1번과 2번 조건을 충족하면서 x 번 도시까지 가는데 걸리는 시간($= \sum_{i=1}^{l-1} w(q_i, q_{i+1})$)이 최소가 되어야 한다.

당신은 1이상 N 이하의 모든 정수 x 에 대해 황혼이 x 번 도시까지 이동할 수 있는지 판별하고, 이동할 수 있다면 황혼이 x 번 도시까지 가는 데 걸리는 시간을 계산해야 한다.

입력

첫 번째 줄에 세 정수 N, M, K 가 공백으로 구분되어 주어진다. ($2 \leq N \leq 2 \times 10^5; 1 \leq M \leq 3 \times 10^5; 0 \leq K \leq \frac{N}{2}$)

$1+i$ 번째 줄에는 오스타니아의 도로를 나타내는 u_i, v_i, w_i 가 공백으로 구분되어 주어진다. 임의의 i 에 대해 u_i 번 도시에서 v_i 번 도시로 가는 도로는 유일하다. ($1 \leq i \leq M; 1 \leq u_i, v_i \leq N; 1 \leq w_i \leq 10^9; w_i$ 는 정수; $u_i \neq v_i$)

$1+M+j$ 번째 줄에는 단순 경로를 나타내는 s_j+1 개의 정수 $s_j, p_{j,1}, \dots, p_{j,s_j}$ 가 공백으로 구분되어 주어진다. K 개의 단순 경로에 대해 각 도시는 최대 1번 나타나며, 입력에 주어진 순서대로 도로를 따라 이동할 수 있음이 보장된다. ($1 \leq j \leq K; s_j \geq 2; \sum_{j=1}^K s_j \leq N$)

출력

공백으로 구분된 N 개의 정수를 첫 번째 줄에 출력한다.

i 번째 정수는 황혼이 모든 조건을 충족하며 i 번 도시까지 가는데 걸리는 시간이어야 한다. 만약 i 번 도시에 도달할 수 없다면 **-1**을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 4 1 1 2 2 1 3 1 2 4 2 3 4 5 2 2 4	0 2 1 6
4 4 2 1 2 2 1 3 1 2 4 2 3 4 5 2 1 3 2 2 4	0 2 -1 -1
11 12 3 1 2 40 2 3 40 3 1 40 2 4 20 3 6 10 9 1 1 10 11 1 3 7 1 7 6 2 6 7 3 7 8 4 4 5 3 4 1 2 4 5 3 3 7 8 2 10 11	0 40 80 60 -1 83 81 90 -1 -1 -1

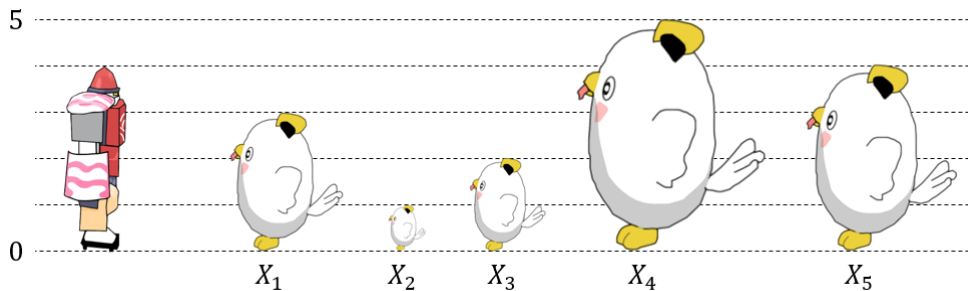
문제 D. 거대 로봇 전투

시간 제한 3 초 메모리 제한 1024 MB



학원 도시 소녀들의 우정과 사랑, 감동의ミリ터리 액션!

블루 아카이브는 2021년 11월 9일에 출시한 수집형 롤플레이нг 게임이다. 블루 아카이브에서는 플레이어가 선생님이 되어 최대 6명의 학생과 함께 PvP, PvE 미션에 참여할 수 있다. 플레이어는 학생에게 특수 스킬을 사용하도록 하여 전략적으로 미션을 수행할 수 있다. 게임 플레이뿐만 아니라 다양한 문제를 해결하는 스토리, 개발사와 유저 간의 활발한 소통으로 2022년 대한민국 게임대상 인기게임상 부문을 수상했다.



거대 로봇과 미니언이 대치하고 있는 모습의 그림이다.

여러분은 선생님이 되어 학생들을 지휘하여 페로로 미니언을 소탕하려고 한다. 학생들은 KAITEN FX Mk.U라는 거대 로봇에 탑승해 임무를 수행한다. 거대 로봇은 제일 왼쪽에 있는 미니언보다 왼쪽에서 임무를 시작하며, 왼쪽으로부터 i ($1 \leq i \leq N$)번째 미니언의 키는 H_i 이고, 거대 로봇의 키는 임무를 시작하기 전에 선생님이 조절할 수 있다. 거대 로봇은 자신이 볼 수 있는 미니언 중 하나를 골라 공격할 수 있다. 거대 로봇이 i 번째 미니언을 볼 수 있으려면, i 번째 미니언 왼쪽에 있는 모든 미니언이 거대 로봇보다 키가 작아야 한다. 거대 로봇이 미니언을 공격하면 해당 미니언은 완전히 사라지며 다른 미니언의 위치나 키는 변하지 않는다.

거대 로봇을 볼 수 있는 미니언들 또한 거대 로봇을 공격한다. i 번째 미니언이 거대 로봇을 볼 수 있으려면, i 번째 미니언 왼쪽에 있는 모든 미니언이 i 번째 미니언보다 키가 작아야 한다. 여러분은 미니언의 공격을 버틸 수 있도록 내구도가 높은 거대 로봇을 준비해야 한다. 내구도가 K 인 거대 로봇으로 임무를 완수하려면 임무 시작부터 끝까지 한 번이라도 K 마리보다 많은 미니언이 거대 로봇을 동시에 공격하면 안 된다.

여러분은 학생들이 타는 거대 로봇의 키에 따라 필요한 내구도가 다르다는 것을 파악했다. 거대 로봇의 키 Q 개가 주어지면 임무를 완수하기 위해 필요한 로봇의 최소 내구도를 구하는 프로그램을 작성하여라.

입력

첫 번째 줄에 미니언의 수 N 과 쿼리의 수 Q 가 공백으로 구분되어 주어진다. ($1 \leq N, Q \leq 250000$)

두 번째 줄에는 각 미니언의 키 H_i 가 공백으로 구분되어 주어진다. ($1 \leq H_i \leq 10^9$)

세 번째 줄부터 Q 개의 줄에는 학생들이 타는 거대 로봇의 키 B_j 가 주어진다. ($1 \leq B_j \leq 10^9$)

입력받는 모든 값은 정수이다.

출력

Q 개의 줄에 걸쳐 거대 로봇의 키가 B_j 일 경우 학생들이 임무를 완수하기 위해 필요한 거대 로봇의 최소 내구도를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 2	2
3 1 2 5 4	3
4	
2	
10 5	2
4 6 1 4 3 1 2 4 5 6	5
7	3
3	5
6	4
2	
4	

문제 E. 격자 속의 직선 경로

시간 제한 2 초 메모리 제한 1024 MB

현대오토에버는 AI, 로봇틱스, IoT, 빅데이터 등 ICT 기술을 기존 생산 공정에 적용하는 스마트 팩토리를 구축하고 있다. 현대오토에버에 다니는 대현이는 새로 구축하는 스마트 팩토리 기술을 적용한 공장의 이동 동선을 최적화하는 일을 하고 있다.

대현이는 가상현실에 실제 공장을 동일하게 구현하는 디지털 트윈 기술을 바탕으로 실제 공장 내에서의 이동을 시뮬레이션해 보고자 한다. 새로 구축하는 공장은 가상 현실 내에서 정사각형 구역 $R \times C$ 개가 R 행 C 열로 붙어 있는 직사각형 격자 형태로 표현된다. 이 중 일부 구역은 이미 설비가 설치되어 있어 내부와 경계를 지나갈 수 없다.

대현이는 1행 1열의 구역에서 생산된 부품을 R 행 C 열의 구역으로 이동시키고자 한다. 직선 방향으로 이동하는 것이 효율적인 만큼, 대현이는 1행 1열 구역 내부 혹은 경계의 어떤 지점에서 R 행 C 열 구역 내부 혹은 경계의 어떤 지점으로 부품을 이동시킬 수 있는 직선 경로가 있는지 궁금하다. 대현이를 도와주자.

입력

첫 번째 줄에 격자의 행과 열의 개수를 나타내는 R 과 C , 그리고 설비가 설치된 구역의 개수 K 가 주어진다. ($2 \leq R, C \leq 1000$; $1 \leq K \leq 100000$)

다음 줄부터 K 개의 줄에 걸쳐서 설비가 설치된 구역의 행 번호 r 과 열 번호 c 가 각 줄에 하나씩 주어진다. ($1 \leq r \leq R$; $1 \leq c \leq C$) 1행 1열이나 R 행 C 열은 주어지지 않으며, 설비가 설치된 구역의 좌표는 서로 다르다.

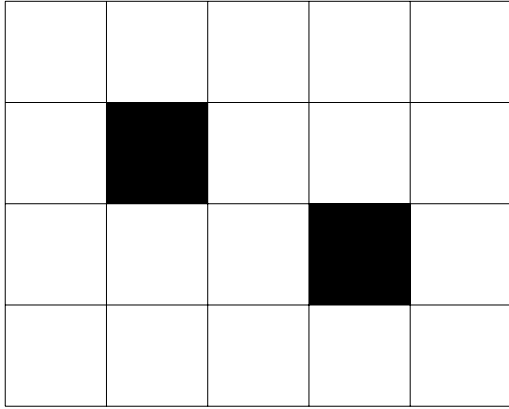
출력

문제의 조건을 만족하는 직선 경로가 있으면 1, 아니면 0을 출력한다.

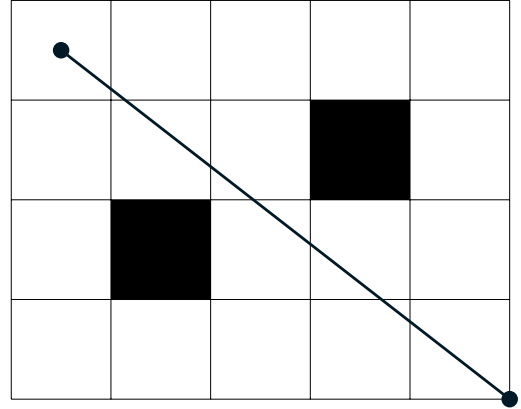
입출력 예시

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

노트



(a) 예제 1에 해당하는 그림



(b) 예제 2에 해당하는 그림

문제 F. 두 트리

시간 제한 5 초 메모리 제한 1024 MB

두 트리 T_1, T_2 와 수열 a_1, a_2, \dots, a_N 이 주어진다. 두 트리는 각각 정점 N 개로 구성되어 있으며, 정점에는 1부터 N 까지의 번호가 겹치지 않게 부여되어 있다. 두 트리의 루트는 1번 정점이다.

T_1 과 T_2 에서 i 번 정점을 루트로 하는 서브트리에 속한 정점의 번호의 집합을 각각 P_i, Q_i 라 하자. $m_i = \max\{a_k \mid k \in P_i \cap Q_i\}$ 로 정의한다.

m_1, m_2, \dots, m_N 을 구하는 프로그램을 작성하시오.

입력

첫 번째 줄에 N 이 주어진다. ($1 \leq N \leq 250000$)

두 번째 줄에 정수 a_1, a_2, \dots, a_N 이 공백으로 구분되어 주어진다. ($1 \leq a_i \leq 10^9$)

다음 $N-1$ 개의 줄에 T_1 의 각 간선이 잇는 두 정점의 번호 u, v 가 공백으로 구분되어 주어진다. ($1 \leq u, v \leq N; u \neq v$)

다음 $N-1$ 개의 줄에 T_2 의 각 간선이 잇는 두 정점의 번호 u, v 가 공백으로 구분되어 주어진다. ($1 \leq u, v \leq N; u \neq v$)

T_1 과 T_2 가 트리 형태임이 보장된다.

출력

m_1, m_2, \dots, m_N 을 차례로 한 줄에 하나씩 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6	100
7 10 5 14 8 100	10
4 5	5
1 5	14
5 6	8
1 2	100
3 6	
1 4	
2 4	
1 5	
3 6	
4 6	

문제 G. K번째 스페이드 찾기

시간 제한 1 초 메모리 제한 1024 MB

유진이는 딜러와 카드 게임을 한다.

딜러는 앞면에 스페이드(♠) 또는 하트(♥) 문양이 그려져 있는 N 장의 카드 묶음을 가지고 있다. 카드는 처음에 뒷면이 보이도록 임의의 순서로 쌓여 있다. 딜러는 모든 카드의 앞면을 알지만, 유진이는 각 카드의 앞면을 모른다.

딜러는 게임을 시작하기 전에 1 이상 N 이하의 정수 K 를 부른다. 유진이의 목표는 위에서 K 번째 스페이드 카드를 찾는 것이다. 이제 다음과 같이 게임이 진행된다.

- 딜러는 카드를 한 장씩 순서대로, 테이블에 뒷면으로 내려놓는다.
- 유진이는 아무 때나 딜러에게 동전 1개를 지불하고 “스탑!”을 외칠 수 있다. (딜러가 마지막 카드를 내려놓은 직후에도 가능하다.)
- 유진이가 “스탑!”을 외친 순간, 딜러는 지금까지 내려놓은 스페이드 카드와 하트 카드의 수를 각각 세어 유진이에게 알려준 뒤 계속해서 카드를 내려놓는다.

딜러가 모든 카드를 내려놓고 나면 딜러는 다시 원래 순서대로 카드를 정리한다. 그다음에 유진이에게 정답을 맞힐 기회가 단 한 번 주어진다. 유진이는 둘 중 하나의 선언을 할 수 있다.

- 카드 묶음에 K 장 미만의 스페이드 카드가 존재한다고 선언한다.
- 카드 한 장을 지목해서, 해당 카드가 위에서 K 번째 스페이드 카드임을 선언한다.

유진이의 선언이 올바르다면 유진이의 승리, 그렇지 않으면 딜러의 승리다.

유진이는 게임을 지는 것을 싫어하기 때문에, 어떤 경우에도 유진이가 승리할 수 있는 필승 전략으로 게임을 진행하려고 한다. 예를 들어, 딜러가 카드를 내려놓을 때마다 “스탑!”을 외치는 전략은 유진이가 반드시 K 번째 카드를 찾을 수 있으므로 필승 전략이다.

유진이는 필승 전략 중에서 가장 적은 동전을 지불하는 전략을 사용하기로 했다. 어떠한 카드 배열이 주어지더라도 다른 필승 전략보다 작거나 같은 양의 동전을 지불해야 하며, 이러한 전략이 존재함을 증명할 수 있다.

처음 카드 배열이 입력으로 주어진다. 1부터 N 까지의 모든 K 값에 대해, 유진이가 딜러에게 지불해야 하는 동전의 개수가 각각 몇 개인지 구하시오.

입력

첫 번째 줄에 카드의 수 N 이 주어진다. ($1 \leq N \leq 10^6$)

두 번째 줄에 카드 묶음을 나타내는 길이 N 의 문자열이 주어진다. i 번째 문자는 위에서 i 번째 카드의 문양을 나타내며, S는 스페이드, H는 하트이다.

출력

한 줄에 N 개의 정수를 공백으로 구분하여 출력한다. i 번째 정수는 $K = i$ 일 때 유진이가 딜러에게 지불해야 하는 동전의 개수이다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
10 HSHHHHHHSH	2 8 5 3 2 1 1 1 1 1
10 SSHHSSHHS	1 1 3 2 5 3 2 1 1 1
10 SSSSSSSSS	1 1 1 1 1 1 1 1 1 1

노트

첫 번째 예제에서, 딜러가 $K=2$ 를 부른 경우의 예시를 보자.

- 유진이는 딜러가 2번째 카드를 내려놓은 뒤 “스탑!”을 외친다. 딜러는 유진이에게 지금까지 하트 1장과 스페이드 1장을 내려놓았다고 알려 준다.
- 이후 유진이는 딜러가 3, 4, ..., 9번째 카드를 내려놓을 때마다 “스탑!”을 외친다. 딜러는 마찬가지로 유진이에게 내려놓은 하트와 스페이드의 개수를 알려준다.
- 딜러가 카드를 정리하고 나면, 유진이는 위에서 9번째 카드가 스페이드 카드임을 맞힐 수 있다.

유진이는 위와 같은 과정으로 8개의 동전을 지불하고 K 번째 스페이드 카드를 찾을 수 있다.

문제 H. 피보나치 반반수열

시간 제한 2 초 메모리 제한 1024 MB

1,2로 시작하는 피보나치 수열 f_n 을 다음과 같이 정의한다.

- $f_1 = 1$
- $f_2 = 2$
- $n \geq 3$ 일 때, $f_n = f_{n-1} + f_{n-2}$

이 피보나치 수열의 반반수열 a_n 을 다음과 같이 정의한다.

- 수열 a_n 은 모든 양의 정수 n 에 대해 $a_n = f_n$ 을 만족하는 모든 수열 중에서 사전 순으로 최소인 수열이다.

수열 a_n 이 b_n 보다 사전 순으로 작다는 것은, 어떤 i 가 존재하여 i 보다 작은 모든 j 에 대해 $a_j = b_j$ 이면서 $a_i < b_i$ 라는 것을 의미한다.

양의 정수 n 이 주어졌을 때, a_n 을 구하시오.

입력

첫 번째 줄에 테스트 케이스의 개수 T 가 주어진다. ($1 \leq T \leq 10000$)

다음 줄부터 T 개의 줄에 걸쳐서 n 의 값이 한 줄에 하나씩 주어진다. ($1 \leq n \leq 10^{18}$)

출력

T 개의 줄에 걸쳐서 각각의 n 에 대해 a_n 의 값을 출력한다. $a_n > 10^{18}$ 이라면 **-1**을 대신 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
7	1
1	2
2	3
3	6
4	13
5	5
6	-1
10000000	

문제 I. 안테나 설치

시간 제한 2 초 메모리 제한 1024 MB

아름다운 도시인 UCPC시에는 총 N 개의 집이 있으며, 각 집은 일직선 상에 동일한 간격으로 배치되어 있다. 이 도시의 시장으로 새로 취임한 도훈이는 시민들의 원활한 네트워크 연결을 위해 신형 안테나를 설치하려고 한다.

세기가 x 인 안테나를 설치하면, 연속하는 l 개의 집에 각각 $x - l + 1$ 만큼의 네트워크 연결 속도가 제공된다. 이때, 네트워크를 제공하는 집의 개수 l 은 x 이하의 정수로 안테나를 설치할 때 직접 설정할 수 있으며, 안테나마다 서로 다른 값으로 설정하는 것도 가능하다. 단, 기술의 한계로 인해 안테나 하나의 최대 세기는 P 이며, 전파 간섭을 막기 위해 두 개 이상의 안테나가 같은 집에 네트워크를 제공하지 않도록 안테나를 설치해야 한다.

모든 시민들을 만족시키기 위해, 도훈이는 각 집에서 요구하는 네트워크 연결 속도를 모두 제공해주려 한다. 또한, 도시의 예산을 아끼기 위해 설치하는 안테나의 세기의 합을 최소화하고자 한다. 도훈이를 위해 각 집에서 요구하는 네트워크 연결 속도를 제공하기 위한 안테나 세기의 합의 최솟값을 구해주자.

입력

첫 번째 줄에 도시에 있는 집의 개수 N 과 안테나 하나의 최대 세기 P 가 공백으로 구분되어 정수로 주어진다. ($1 \leq N \leq 500000$; $1 \leq P \leq 10^9$)

두 번째 줄에 각 집에서 요구하는 네트워크 연결 속도를 나타내는 정수 a_i 가 공백으로 구분되어 순서대로 주어진다. ($1 \leq a_i \leq P$)

출력

모든 집에서 요구하는 네트워크 연결 속도를 제공하기 위한 안테나 세기의 합의 최솟값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
7 5 2 4 3 3 1 4 5	19

노트

1번째 집부터 2번째 집까지 4의 연결 속도를 제공하는 세기 5의 안테나를, 3번째 집부터 5번째 집까지 3의 연결 속도를 제공하는 세기 5의 안테나를, 6번째 집에 4의 연결 속도를 제공하는 세기 4의 안테나를, 7번째 집에 5의 연결 속도를 제공하는 세기 5의 안테나를 설치하면 안테나 세기의 합은 19로 모든 집의 요구사항을 만족시킬 수 있다.

문제 J. 회전초밥

시간 제한 2 초 메모리 제한 1024 MB

UCPC시에는 세계 최고로 유명한 회전초밥집이 있다. 그 이름은 바로 "뭐든지 회전하는 회전초밥집"이다! 이 회전초밥집은 요리사도, 손님들도 전부 회전하는 독특한 컨셉으로 SNS에서 큰 인기를 끌었다. 이 유명한 가게를 놓칠 수 없었던 UCPC 출제진들은 몇 달 전에 가게에 예약하였고, 오늘 드디어 그 회전초밥집에 방문하게 되었다!

총 N 명의 출제진이 회전초밥집을 방문하였고, 주방에서는 이에 맞추어 $N+1$ 명의 요리사를 대기시켰다. i 번째 요리사는 1분마다 b_i 개의 초밥을 만들 수 있다. 처음에 웨이터가 출제자들을 일렬로 앉힌 뒤, i 번째 위치한 출제자에게 a_i 개의 초밥을 나누어주었다. 그리고 1분마다 다음의 행동이 순서대로 반복되었다.

1. $1 \leq i \leq N$ 을 만족하는 모든 i 에 대하여, i 번째에 위치한 요리사가 i 번째에 위치한 출제자에게 초밥을 만들어준다. 이때, 만들어주는 초밥의 양은 그 요리사가 1분마다 만들 수 있는 초밥의 양과 같다. $N+1$ 번째에 위치한 요리사는 초밥을 만들지 않고 휴식한다.
2. 요리사들이 한 번 회전한다. 즉, 원래 1번째에 위치한 요리사는 2번째로, 원래 2번째에 위치한 요리사는 3번째로..., 원래 $N+1$ 번째에 위치한 요리사는 1번째로 이동한다.
3. 출제진이 한 번 회전한다. 즉, 원래 1번째에 위치한 출제자는 2번째로, 원래 2번째에 위치한 출제자는 3번째로..., 원래 N 번째에 위치한 출제자는 1번째로 이동한다.

각 출제자는 언제라도 자신에게 초밥 한 세트가 모여있다면 바로 그 초밥 세트를 먹는다. 여기서 초밥 한 세트는 종류 상관없이 초밥 K 개의 묶음을 의미한다. 초밥을 먹는 데 걸리는 시간은 무시한다.

출제자들은 음식을 남기는 것을 아주 싫어하기 때문에, 모두가 가지고 있는 초밥의 양이 0이 될 때까지 식사하려고 한다. 식사를 시작한 지 몇 분 후에 식사를 끝마치게 될까?

입력

첫 번째 줄에 정수 N, K 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 2000; 2 \leq K \leq 1000000$)

두 번째 줄에 a_1, a_2, \dots, a_N 이 공백으로 구분되어 주어진다. ($0 \leq a_i \leq K-1$)

세 번째 줄에 b_1, b_2, \dots, b_{N+1} 이 공백으로 구분되어 주어진다. ($1 \leq b_i \leq K-1$)

출력

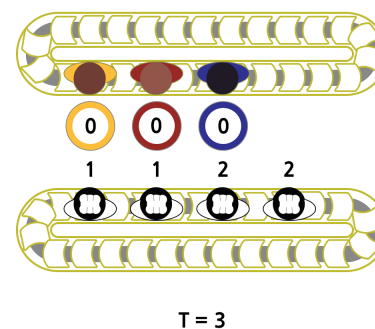
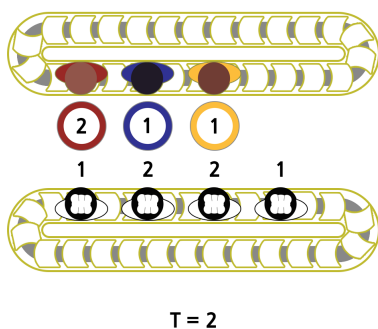
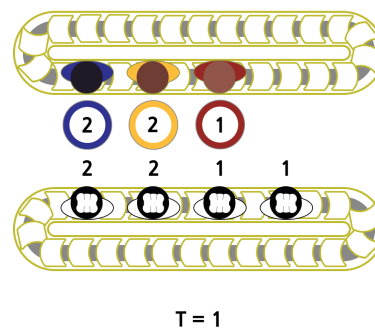
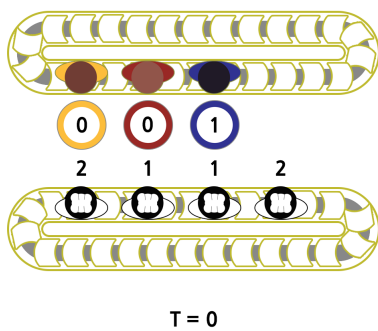
출제진이 식사를 끝마칠 때까지 걸린 시간을 분 단위로 출력한다. 만약 무한한 시간이 지나도 출제진이 식사를 끝마치지 못한다면 대신 **-1**을 출력한다.

입출력 예시

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

노트

첫 번째 예제의 경우, 각 시간대 별로 출제진, 요리사의 위치 및 각 출제자가 받은 초밥의 양, 요리사가 만들 수 있는 초밥의 양을 시각화하면 다음과 같다.



두 번째 예제의 경우,

문제 K. 그건 가지가 아니라 대파예요

시간 제한 2 초 메모리 제한 1024 MB

어버진과 스칼리온이 N 행 M 열 격자 모양 게임판에서 게임을 한다. 격자의 각 칸에는 ‘가’, ‘지’, ‘대’, ‘파’, ‘☆’ 중 하나가 적혀있다.

이 게임에서는 스칼리온과 어버진이 번갈아 자기 턴을 수행한다. 스칼리온이 먼저 턴을 수행한다.

스칼리온은 자신의 차례에서 왼쪽 칸이 ‘가’이고 오른쪽 칸이 ‘지’인 가로 방향으로 인접한 두 타일을 선택하여 해당 두 칸을 ‘대’, ‘파’로 바꿀 수 있다. 어버진은 자신의 차례에서 위쪽 칸이 ‘대’이고 아래쪽 칸이 ‘파’인 세로 방향으로 인접한 두 타일을 선택하여 해당 두 칸을 ‘가’, ‘지’로 바꿀 수 있다.

어버진과 스칼리온 모두 타일을 바꾸는 방법이 있다면 반드시 타일을 바꿔야 한다. 게임을 진행하다가 자기 턴을 수행할 수 없는 쪽은 지게 된다. 만약 10^{100} 턴이 지난 이후에도 게임이 끝나지 않는다면 게임이 무승부로 종료된다.

어버진과 스칼리온은 각자 최선을 다하여 게임을 플레이한다. 즉, 두 플레이어는 이기는 방법이 있으면 이기는 수를 선택하며 이기는 방법이 없지만 비기는 방법이 있으면 비기는 수를 선택한다.

게임판이 주어지면 게임의 승자를 구하는 프로그램을 작성하여라.

입력

첫 번째 줄에 게임판의 크기 N, M ($2 \leq N, M \leq 2500$) 이 공백으로 구분되어 주어진다.

두 번째 줄부터 N 개의 줄에 각 격자의 내용이 알파벳 대문자로 공백 없이 주어진다. ‘G’는 ‘가’, ‘Z’는 ‘지’, ‘D’는 ‘대’, ‘P’는 ‘파’, ‘X’는 ‘☆’에 대응된다.

출력

어버진과 스칼리온이 최선을 다하여 게임을 했을 때 어버진이 이긴다면 **Aubergine**을, 스칼리온이 이긴다면 **Scallion**을, 무승부라면 **Draw**를 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
3 3 GZX PXG XGZ	Scallion
2 2 GZ PD	Aubergine

문제 L. 나무늘보

시간 제한 2 초 메모리 제한 1024 MB

N 개의 정점으로 구성된 이진트리에 거주하고 있는 나무늘보가 있다. 각 정점에는 1부터 N 까지 번호가 적혀 있다. 나무늘보는 현재 거주지를 신고하기 위해 거주하고 있는 이진트리의 전위, 후위 순회 결과를 적어 두었다. 이진트리의 전위, 후위 순회의 정의는 아래와 같다.

- 전위 순회: 현재 정점을 방문한 후 왼쪽, 오른쪽 순서로 서브 트리를 전위 순회한다.
- 후위 순회: 왼쪽, 오른쪽 순서로 서브 트리를 후위 순회한 후 현재 정점을 방문한다.

나무늘보는 적어 둔 전위, 후위 순회 결과로 이루어진 이진트리가 여러 개 존재할 수도 있고, 결과를 잘못 적어서 이진트리가 존재하지 않을 수도 있다는 사실을 알게 되었다.

당신은 나무늘보가 적어 둔 전위, 후위 순회 결과가 주어졌을 때 동일한 순회 결과를 가지는 이진트리의 개수를 알아내야 한다.

입력

첫 번째 줄에 이진트리의 정점 개수 N 이 주어진다. ($1 \leq N \leq 500000$)

두 번째 줄에는 전위 순회 결과를 나타내는 정수 N 개가 공백으로 구분되어 주어진다.

세 번째 줄에는 후위 순회 결과를 나타내는 정수 N 개가 공백으로 구분되어 주어진다.

입력으로 들어오는 전위, 후위 순회 결과는 1부터 N 까지의 정수가 중복 없이 배치된 순열이다.

출력

문제의 답을 998244353으로 나눈 나머지를 한 줄에 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5 4 1 3 5 2 1 5 2 3 4	1
7 1 2 3 4 5 6 7 4 5 3 2 7 6 1	4
2 1 2 1 2	0

문제 M. 산 색칠

시간 제한 2 초 메모리 제한 1024 MB

히스토그램 모양의 도화지에 산 색칠을 반복하여 히스토그램 모양의 그림을 그리려고 한다.

히스토그램은 밑변이 x 축에 평행한 한 개 이상의 직사각형이 공통된 밑변을 가지면서 빈틈없이 좌우로 붙어있는 도형을 말한다. 각 직사각형의 너비와 높이는 정수이며 서로 다를 수 있다.

한 번의 산 색칠은 다음과 같이 이루어진다.

- 도화지의 윗변 중 한 곳을 산의 정상으로 잡는다.
- 그에 맞는 산 모양 중 넓이가 **최대**인 모양으로 색칠한다.

산은 도화지를 벗어날 수 없으며, 이전의 색칠은 현재 색칠에 영향을 주지 않는다. 또한 직사각형의 왼쪽이나 오른쪽 끝은 산의 정상이 될 수 없다.

산의 구체적인 정의는 다음과 같다.

- 산의 정상이 가장 높은 히스토그램 모양이다.
- 산의 정상을 포함하는 왼쪽 부분은 높이가 단조 증가한다.
- 산의 정상을 포함하는 오른쪽 부분은 높이가 단조 감소한다.

단, 함수 f 가 $x_1 < x_2$ 인 모든 x_1, x_2 에 대하여 $f(x_1) \leq f(x_2)$ 을 만족하면 단조 증가한다고 하고, $f(x_1) \geq f(x_2)$ 을 만족하면 단조 감소한다고 한다.

산 색칠을 반복하여 주어진 그림을 그릴 수 있다면 최소한의 색칠 횟수로 그리고, 불가능하다면 **-1**을 출력하시오.

입력

첫 번째 줄에 도화지의 히스토그램을 이루는 직사각형의 개수 N 이 주어진다. ($1 \leq N \leq 200000$)

두 번째 줄부터 N 개의 줄에 걸쳐 각 직사각형의 너비와 높이 w_i, h_i 가 공백으로 구분되어 주어진다. ($1 \leq w_i, h_i \leq 10^9$)

그 다음 줄에 그림의 히스토그램을 이루는 직사각형의 개수 M 이 주어진다. ($1 \leq M \leq 200000$)

그 다음 줄부터 M 개의 줄에 걸쳐 각 직사각형의 너비와 높이 w_j, h_j 가 공백으로 구분되어 주어진다. ($1 \leq w_j, h_j \leq 10^9$)

- 도화지와 그림의 총 너비는 같으며 10^9 을 넘지 않는다. ($\sum w_i = \sum w_j \leq 10^9$)
- 도화지는 그림을 포함한다.
- 인접한 직사각형은 높이가 서로 다르다. ($h_i \neq h_{i+1}; h_j \neq h_{j+1}$)

출력

첫 번째 줄에 필요한 산 색칠의 최소 횟수를 출력한다. 만일 그림을 완성할 수 없다면 **-1**을 출력하고 종료한다.

두 번째 줄에 각 색칠에서 산의 정상이 도화지의 몇 번째 직사각형에 포함되는지 공백으로 구분하여 출력한다.

정답이 여러 가지라면 그중 아무거나 출력한다.

입출력 예시

표준 입력(stdin)

표준 출력(stdout)

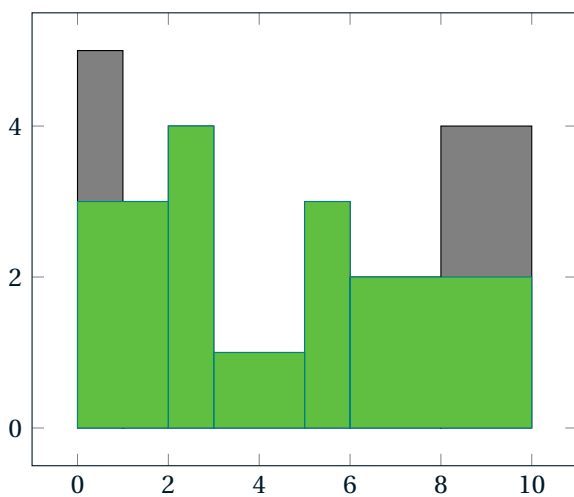
```
7
1 5
1 3
1 4
2 1
1 3
2 2
2 4
5
2 3
1 4
2 1
1 3
4 2
```

```
2
5 3
```

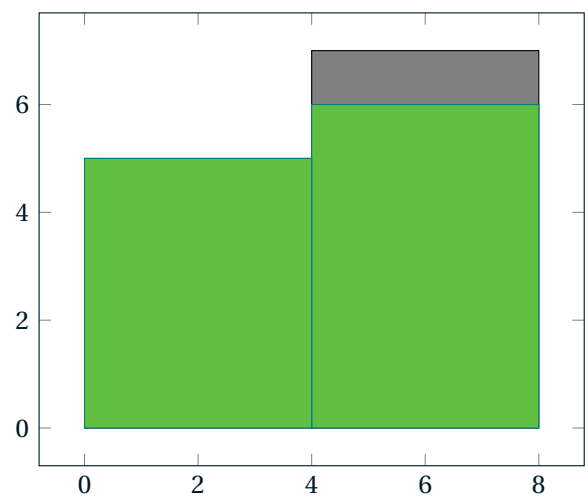
```
2
4 5
4 7
2
4 5
4 6
```

```
-1
```

노트



(a) 첫 번째 예제의 도화지와 그림



(b) 두 번째 예제의 도화지와 그림