



전국 대학생 프로그래밍 대회 동아리 연합
여름 대회 2026

Preliminaries

Official Problemset

예선 문제

2026년 6월 27일, 14:00 - 17:00

주최

전국 대학생 프로그래밍 대회 동아리 연합

후원



ASTEROMORPH



김동현(kdh9949) 김영현(kipa00) 나정휘(jhnah917) 정재호(cubic)

문제 목록

문제지에 있는 문제가 총 **12**문제가 맞는지 확인하시기 바랍니다.

A 체육은 수학과목입니다 3

B Bakejoon Online Judge

C Hat Trick

D 괄호 게임

E 공항

F 괄호와 쿼리

G 레이저 타워

H 빈 상자

I 논문 공장

J 어려운 정렬 문제

K 상인

L 확률성 정렬

모든 문제의 메모리 제한은 1024 MB로 동일합니다.

문제 A. 체육은 수학과목입니다 3

시간 제한 1 초 메모리 제한 1024 MB

“지각이야, 지각!”



생성형 AI로 만든 참고 이미지

UCPC 초등학교에 다니는 민철이는 오늘 아침 시계를 보고 깜짝 놀랐다. 시계를 보니 이미 오전 8시가 넘어 있었기 때문이다.

민철이는 서둘러 집을 나와 학교 정문까지 뛰어가려고 한다. UCPC 초등학교는 오전 9시가 되는 순간 무서운 선생님이 학교 정문을 꽁 닫는다. 지각 여부는 민철이가 학교 정문에 도착하는 시각을 기준으로 판단한다. 따라서 정확히 오전 9시에 정문에 도착해도 지각이며, 그보다 먼저 정문에 도착해야 한다.

UCPC 초등학교에 다니는 학생들은 지각하면 반성문을 써야 한다. 민철이는 반성문을 쓰기 싫어서 최대한 빠르게 달리려고 한다. 하지만 아무리 열심히 뛰어도 지각할 것 같다면, 어차피 써야 할 반성문이니 조금 더 늦더라도 근처 떡볶이집에 들러 떡볶이를 먹고 갈 생각이다.

학교 정문까지 남은 거리, 현재 시각, 그리고 민철이가 뛰여가는 속도가 주어진다. 체력이 좋은 민철이는 학교 정문에 도착할 때까지 일정한 속도로 달릴 수 있다. 민철이가 지금 바로 뛰여갔을 때 지각하지 않을 수 있는지 알려주자.

입력

첫째 줄에 학교 정문까지 남은 거리 D 미터, 현재 시각이 오전 8시 M 분 0초일 때의 M , 민철이가 뛰여가는 속도 V 가 공백을 사이에 두고 주어진다. ($1 \leq D \leq 10000$; $1 \leq M \leq 59$; $50 \leq V \leq 300$)

민철이의 속도는 분당 V 미터이다.

입력으로 주어지는 모든 수는 정수임이 보장된다.

출력

민철이가 오전 9시가 되기 전에 학교 정문에 도착할 수 있다면 run을 출력한다.

그렇지 않다면 late를 출력한다.

입출력 예시

stdin	stdout
300 55 100	run
700 55 100	late
600 50 60	late

노트

첫 번째 예제의 경우 현재 시각은 오전 8시 55분이며, 학교 정문이 닫히기까지 5분이 남아 있다.

민철이는 분당 100미터의 속도로 달릴 수 있으므로, 300미터를 이동하는 데 3분이 걸린다. 따라서 오전 8시 58분에 학교 정문에 도착할 수 있으므로 지각하지 않는다.

두 번째 예제의 경우 현재 시각은 오전 8시 55분이며, 학교 정문이 닫히기까지 5분이 남아 있다.

민철이는 분당 100미터의 속도로 달릴 수 있으므로, 700미터를 이동하는 데 7분이 걸린다. 따라서 오전 9시가 되기 전에 학교 정문에 도착할 수 없으므로 지각한다.

세 번째 예제의 경우 현재 시각은 오전 8시 50분이며, 학교 정문이 닫히기까지 10분이 남아 있다.

민철이는 분당 60미터의 속도로 달릴 수 있으므로, 600미터를 이동하는 데 정확히 10분이 걸린다. 따라서 오전 9시 정각에 학교 정문에 도착하게 된다. 문제의 조건에 따라 오전 9시에 도착하는 경우도 지각이므로 정답은 late이다.

문제 B. Bakejoon Online Judge

시간 제한 2 초 메모리 제한 1024 MB

Bakejoon Online Judge는 특이한 조건을 만족하는 빵을 구워서 제출하는 온라인 저지이다.

Bakejoon Online Judge의 유저인 당신은 다음 조건을 만족하는 N 개의 신선한 빵을 한 번에 제출해야 한다.

- 빵은 한 번에 최대 한 개만 구울 수 있으며, 굽기 시작한 빵을 중단할 수는 없다.
- 각 빵은 준비 시간 P_i , 굽는 시간 T_i , 신선도 F_i 라는 특성을 가진다.
 - 빵을 굽기 전의 준비 시간이 필요하므로 i 번째 빵은 시각 P_i 부터 구울 수 있다.
 - i 번째 빵을 굽는 데에는 T_i 시간이 걸린다.
 - i 번째 빵은 완전히 구워진 시각부터 F_i 시간 동안 신선함이 유지된다.

엄밀히는, 시각 $t \geq P_i$ 에 i 번째 빵을 굽기 시작하면 시각 $t+T_i$ 에 빵이 완전히 구워지고, 시각 $t+T_i$ 부터 시각 $t+T_i+F_i$ 까지 신선함이 유지된다. 시각이 정확히 $t+T_i$ 또는 $t+T_i+F_i$ 일 때도 빵은 신선하다.

N 개의 빵의 특성이 주어질 때 모든 빵을 신선한 상태로 제출할 수 있는지와 가능하다면 빵을 제출할 수 있는 가장 이른 시각을 구해보자.

입력

첫째 줄에 빵의 개수 N 이 주어진다. ($1 \leq N \leq 200\,000$)

둘째 줄부터 N 개의 줄에 걸쳐 각 빵의 준비 시간 P_i , 굽는 시간 T_i , 신선도 F_i 가 공백을 사이에 두고 주어진다. ($0 \leq P_i, T_i, F_i \leq 10^9$)

출력

N 개의 빵을 모두 신선한 상태로 제출할 수 있는 가장 이른 시각을 출력한다. 만약 N 개의 빵을 모두 신선한 상태로 제출할 수 없다면 -1을 대신 출력한다.

입출력 예시

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

노트

첫 번째 예제의 경우 다음 방식대로 빵을 구우면 시각 7에 모든 빵을 신선한 상태로 제출할 수 있다.

0	1	2	3	4	5	6	7	8
X	X	P	P	P	T	F	F	F
X	X	X	X	T	F	F	R	
X	P	P	P	P	T	T	F	R
T	T	T	F	F	F	F	F	F

제출!

그림에서 회색 X는 아직 빵을 구울 준비가 되지 않았음을, 초록색 P는 준비가 완료되었음을, 빨간색 T는 오븐에서 빵을 굽고 있음을, 파란색 F는 빵이 신선한 상태임을, 노란색 R은 빵이 더 이상 신선하지 않음을 나타낸다.

두 번째 예제의 경우 어떻게 빵을 굽더라도 3개의 빵을 모두 신선한 상태로 만들 수 없다.

문제 C. Hat Trick

시간 제한 0.25 초 메모리 제한 1024 MB

이 문제는 두 스텝 문제입니다.

브루는 마술 쇼를 관람하던 중 다음과 같은 마술을 보게 되었다.

- 조수는 총 $2N$ 개의 모자를 준비하는데, 이 중 N 개는 빨간색이고, N 개는 파란색이다.
- 조수는 N 명의 관객에게 차례로 다가가, 1번부터 N 번 관객의 순서대로 남은 모자 중 원하는 색의 모자 하나를 선택해 가져가도록 한다. 마술사는 이 과정을 보지 않는다.
- 그 뒤 조수는 N 개의 남은 모자들을 긴 탁자 위에 일렬로 배치한다.
- 마술사는 탁자 위에 놓인 모자들만을 보고, 각 관객이 가져간 모자의 색을 맞힌다.

마술이 시작되기 전, 조수와 마술사는 어떤 방식으로 행동할지 미리 약속해 둘 수 있다. 하지만 마술이 시작된 뒤 조수가 마술사에게 정보를 줄 수 있는 방법은 탁자 위에 모자를 놓는 것뿐이다.

마술을 성공시키기 위한 조수와 마술사의 전략을 직접 구현해 보자.

이 문제에서 당신의 프로그램은 한 테스트 데이터에 대해 두 번 실행된다. 첫 번째 실행에서는 조수의 역할을 수행해야 하고, 두 번째 실행에서는 마술사의 역할을 수행해야 한다.

입력

첫 번째 줄에는 이번 실행에서의 역할을 나타내는 문자열 R 이 주어진다. R 은 assistant 또는 magician 중 하나이며, 각각 이번 실행에서의 역할이 조수 또는 마술사임을 의미한다.

두 번째 줄에는 관객의 수 N 이 주어진다. ($1 \leq N \leq 100$)

R 이 assistant인 경우, 세 번째 줄에는 길이 N 의 문자열 C 가 주어진다. C_i 는 i 번 관객이 빨간 모자를 가져갔다면 R, 파란 모자를 가져갔다면 B이다.

R 이 magician인 경우, 세 번째 줄에는 길이 N 의 문자열 D 가 주어진다. D_i 는 조수가 왼쪽에서부터 i 번째 위치에 놓은 모자가 빨간색이라면 R, 파란색이라면 B이다. 이 문자열은 같은 테스트의 첫 번째 실행에서 조수 역할의 프로그램이 출력한 문자열과 동일하다.

출력

R 이 assistant인 경우, 첫 번째 줄에 조수가 놓은 N 개의 모자의 색을 왼쪽부터 차례대로 공백 없이 출력한다. 빨간 모자는 R, 파란 모자는 B로 출력한다. 이때 조수가 놓은 모자와 관객들이 가져간 모자를 합하면 빨간 모자와 파란 모자가 각각 정확히 N 개씩 존재해야 한다.

R 이 magician인 경우, 첫 번째 줄에 마술사가 추측한 각 관객이 가져간 모자의 색을 1번 관객부터 N 번 관객까지 차례대로 공백 없이 출력한다. 빨간 모자는 R, 파란 모자는 B로 출력한다. 이때의 출력은 첫 번째 실행에서 주어진 C 와 정확히 같아야 한다.

Preliminaries - 예선대회

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
assistant 4 RBBR	BRBR
magician 4 BRBR	RBBR

노트

위의 두 예시는 각각 하나의 테스트 데이터에서 첫 번째 실행과 두 번째 실행을 나타낸다. 첫 번째 실행에서 출력된 문자열 BRBR이 두 번째 실행의 입력 문자열과 같음을 유의하라.

문제 D. 괄호 게임

시간 제한 1 초 메모리 제한 1024 MB

짝수인 양의 정수 N 과 정수 $A_1, A_2, \dots, A_N$ 이 주어진다. Alice와 Bob이 괄호 게임을 한다. 게임은 다음과 같이 진행된다.

1. Alice가 $A_1, A_2, \dots, A_N$ 을 원하는 순서대로 재배열한다.
2. Bob은 재배열된 A 를 보고 길이가 N 인 올바른 괄호 문자열 S 를 고른다. (올바른 괄호 문자열의 정의는 지문 하단부에서 확인할 수 있다.)

게임의 점수는 $S_i = ($ 인 A_i 들의 합에서 $S_i =)$ 인 A_i 들의 합을 뺀 값이다. Alice는 점수를 최대화, Bob은 점수를 최소화하려고 한다. Alice와 Bob 모두 최적의 전략을 사용할 때, 게임의 최종 점수를 구하여라.

입력

첫 번째 줄에 정수 N 이 주어진다. ($2 \leq N \leq 200\,000$; N 은 짝수)

두 번째 줄에 N 개의 정수 $A_1, A_2, \dots, A_N$ 이 공백으로 구분되어 주어진다. ($1 \leq A_i \leq 10^9$)

출력

두 플레이어가 최적의 전략을 사용할 때, 게임의 최종 점수를 출력한다.

입출력 예시

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

노트

올바른 괄호 문자열의 정의는 다음과 같다.

- 빈 문자열은 올바른 괄호 문자열이다.
- 두 올바른 괄호 문자열 S 와 T 에 대해, 두 문자열을 이어붙인 ST 도 올바른 괄호 문자열이다.
- 올바른 괄호 문자열 S 에 대해, S 를 괄호로 둘러싼 (S) 도 올바른 괄호 문자열이다.
- 위 방식으로 만들 수 없는 문자열은 올바른 괄호 문자열이 아니다.

문제 E. 공항

시간 제한 1 초 메모리 제한 1024 MB

N 개의 도시가 있다. 하지만 현재 교통 시설이 지어지지 않아, 도시 사이를 오갈 수 없다.

도로 건설 계획이 M 개 있다. i 번째 계획을 선택하면 도시 u_i 와 도시 v_i 를 잇는 양방향 도로를 w_i 의 비용으로 건설할 수 있다.

또한, 각 도시에는 공항을 지을 수 있다. 도시 i 에 공항을 짓는 비용은 A_i 이다. 공항이 지어진 두 도시 사이는 비행기를 이용하여 자유롭게 이동할 수 있다.

도로 건설과 공항 건설 중 일부를 선택하여, 어떤 두 도시를 골라도 도로와 비행기를 이용하여 한 도시에서 다른 도시로 이동할 수 있게 하려고 한다. 이때, 조건을 만족시키기 위해 필요한 총 건설 비용의 최솟값을 구하여라.

입력

첫째 줄에 도시의 수 N 과 도로 건설 계획의 수 M 이 공백을 사이에 두고 주어진다. ($1 \leq N \leq 10^5$; $0 \leq M \leq 10^5$)

다음 M 개의 줄에 걸쳐 각 계획의 정보가 주어진다.

이 중 i 번째 줄에는 세 정수 u_i, v_i, w_i 가 공백을 사이에 두고 주어진다. 이는 도시 u_i 와 도시 v_i 를 잇는 양방향 도로를 w_i 의 비용으로 건설할 수 있음을 의미한다. ($1 \leq u_i, v_i \leq N$; $u_i \neq v_i$; $1 \leq w_i \leq 10^9$)

마지막 줄에 N 개의 정수 $A_1, A_2, \dots, A_N$ 이 공백을 사이에 두고 주어진다. A_i 는 도시 i 에 공항을 짓는 비용이다. ($1 \leq A_i \leq 10^9$)

여러 개의 도로 건설 계획이 같은 도시 쌍 사이를 이을 수 있다.

출력

어떤 두 도시 사이도 이동 가능하도록 만드는 데 필요한 총 건설 비용의 최솟값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
4 2 1 2 4 3 4 4 3 10 2 10	13
3 3 1 2 1 2 3 1 1 3 10 100 100 100	2

노트

첫 번째 예제에서는 도시 1과 도시 2를 잇는 도로, 도시 3과 도시 4를 잇는 도로를 건설하고, 도시 1과 도시 3에 공항을 짓는 것이 최적이다. 총 비용은 $4+4+3+2=13$ 이다.

Preliminaries - 예선대회

두 번째 예제에서는 공항을 짓지 않고 도시 1과 도시 2, 도시 2와 도시 3을 잇는 도로를 건설하는 것이 최적이다. 총 비용은 $1 + 1 = 2$ 이다.

문제 F. 괄호와 쿼리

시간 제한 5 초 메모리 제한 1024 MB

길이가 N 인 두 수열 a, b 가 주어진다. 당신은 다음과 같은 Q 개의 쿼리를 처리해야 한다.

- $l\ r\ k$: 구간 $[l, r]$ 의 각 위치 i 를 (또는) 로 정해 올바른 괄호문자열을 만든다. (올바른 괄호 문자열의 정의는 지문 하단부에서 확인할 수 있다.) 이 때 위치 i 의 괄호가 (라면 a_i 점을,)라면 b_i 점을 얻는다. 총점이 최대가 되도록 괄호문자열을 정하는 모든 방법에 대해 위치 k 에 있는 괄호가 동일하다면 해당 괄호의 방향을 출력한다. 괄호의 방향이 하나로 정해지지 않는다면 x 를 출력한다.

입력

첫째 줄에 두 정수 N, Q 가 공백으로 구분되어 주어진다. ($1 \leq N, Q \leq 500000$)

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

셋째 줄에 N 개의 정수 $b_1, b_2, \dots, b_N$ 이 공백으로 구분되어 주어진다. ($1 \leq b_i \leq 10^9$)

다음 Q 개의 줄에 걸쳐 각 쿼리를 나타내는 3개의 정수 l, r, k 가 공백으로 구분되어 주어진다. ($1 \leq l \leq k \leq r \leq N$; $r - l + 1$ 은 짝수)

출력

Q 개의 줄에 걸쳐 각 쿼리의 답을 출력한다.

입출력 예시

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

노트

올바른 괄호 문자열의 정의는 다음과 같다.

- 빈 문자열은 올바른 괄호 문자열이다.
- 두 올바른 괄호 문자열 S 와 T 에 대해, 두 문자열을 이어붙인 ST 도 올바른 괄호 문자열이다.
- 올바른 괄호 문자열 S 에 대해, S 를 괄호로 둘러싼 (S) 도 올바른 괄호 문자열이다.
- 위의 방식으로 만들 수 없는 문자열은 올바른 괄호 문자열이 아니다.

문제 G. 레이저 타워

시간 제한 2 초 메모리 제한 1024 MB

수평선 위의 $x=1, 2, \dots, N$ 에 각각 타워가 하나씩 있다. 위치가 $x=i$ 인 타워의 높이는 h_i 이며, $h_1, h_2, \dots, h_N$ 은 1부터 N 까지의 정수로 이루어진 순열이다.

각 타워의 꼭대기인 (i, h_i) 에서 레이저를 반반의 확률로 왼쪽 또는 오른쪽으로 발사한다.

- 레이저를 왼쪽으로 발사하는 경우 레이저는 $0 \leq x \leq i$ 이고 $0 \leq y \leq h_i$ 인 직사각형 영역을 덮는다.
- 레이저를 오른쪽으로 발사하는 경우 레이저는 $i \leq x \leq N+1$ 이고 $0 \leq y \leq h_i$ 인 직사각형 영역을 덮는다.

모든 타워의 레이저 발사 방향은 상호 독립적으로 결정된다.

모든 타워에서 레이저를 발사했을 때 덮인 영역들의 합집합의 넓이를 S 라고 하자. 2^S 의 기댓값을 998244353으로 나눈 나머지를 구하여라.

입력

첫째 줄에 타워의 개수 N 이 주어진다. ($2 \leq N \leq 200000$)

둘째 줄에 각 타워의 높이를 의미하는 N 개의 정수 $h_1, h_2, \dots, h_N$ 이 공백으로 구분되어 주어진다. 주어지는 높이는 1부터 N 까지의 정수가 한 번씩 등장하는 순열이다.

출력

2^S 의 기댓값을 998244353으로 나눈 나머지를 출력한다. 998244353은 소수이다.

구체적으로, 기댓값은 항상 유리수임이 보장되며, 이를 기약분수 P/Q 로 나타내었을 때 $Q \not\equiv 0 \pmod{998244353}$ 임을 보일 수 있다. 이때, $P \times Q^{-1} \pmod{998244353}$ 의 값을 출력해야 한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2 1 2	16
3 1 2 3	768
6 6 5 2 3 4 1	448265500

노트

예제 1에서 전체 공간의 범위는 $x=0$ 부터 $x=3$ 까지이다. 1번 타워와 2번 타워가 레이저를 쏘는 4가지 방향 조합에 대해, 덮인 영역들의 합집합의 넓이 S 와 2^S 의 값은 다음과 같다. 각각의 상황은 $1/4$ 의 확률로 일어난다.

- 왼쪽, 왼쪽: $0 \leq x \leq 2$ 구간이 높이 2로 덮인다. ($S=4$, $2^S=16$)

Preliminaries - 예선대회

- 왼쪽, 오른쪽: $0 \leq x \leq 1$ 은 높이 1, $2 \leq x \leq 3$ 은 높이 2로 덮인다. ($S = 3$, $2^S = 8$)
- 오른쪽, 왼쪽: $0 \leq x \leq 2$ 는 높이 2, $2 \leq x \leq 3$ 은 높이 1로 덮인다. ($S = 5$, $2^S = 32$)
- 오른쪽, 오른쪽: $1 \leq x \leq 2$ 는 높이 1, $2 \leq x \leq 3$ 은 높이 2로 덮인다. ($S = 3$, $2^S = 8$)

따라서 2^S 의 기댓값은 $\frac{16+8+32+8}{4} = 16$ 이다.

문제 H. 빈 상자

시간 제한 1 초 메모리 제한 1024 MB

니나에게는 용량이 X 인 빈 상자가 있다. 이 빈 상자는 유난히 길어서, 값을 앞쪽과 뒤쪽 중 원하는 쪽에 넣을 수 있다. 또한 이 상자는 다음과 같은 특별한 성질을 가진다.

어떤 값을 넣었을 때 상자 안에 들어 있는 값의 개수가 X 를 초과하면, 즉 $X+1$ 개가 되면, 상자 안에 들어 있는 값들 중 가장 작은 값이 제거된다. 가장 작은 값이 여러 개라면, 그중 가장 앞에 있는 값이 제거된다.

니나는 상자의 정확한 용량 X 를 모른다. 대신 $1 \leq X \leq P$ 라는 정보가 추가로 주어진다고 하자. 다음 과정을 통해 어떤 경우에도 X 의 값을 $1, 2, \dots, P$ 중 하나로 확인할 수 있다면, 이 P 는 가능하다고 한다.

다시 말해, 서로 다른 두 용량 $X_1, X_2 \in \{1, 2, \dots, P\}$ 에 대해 관찰되는 결과가 항상 다르다면 P 는 가능하다.

- 길이 N 의 정수열 $A_1, A_2, \dots, A_N$ 과 길이 N 의 이진 수열 $B_1, B_2, \dots, B_N$ 이 주어진다.
- 상자는 최초에 비어 있으며, $i = 1, 2, \dots, N$ 에 대해 다음 시행을 순서대로 한다.
 - $B_i = 0$ 이면 A_i 를 상자의 앞에 넣는다.
 - $B_i = 1$ 이면 A_i 를 상자의 뒤에 넣는다.
 - 필요하다면 가장 작은 값을 제거한다.
 - 그 후, 상자의 맨 앞에 있는 값을 확인한다.

가능한 P 중 최댓값을 구하여라.

입력

첫째 줄에 정수 N 이 주어진다. ($1 \leq N \leq 10^5$)

둘째 줄에 N 개의 정수 $A_1, A_2, \dots, A_N$ 이 공백을 사이에 두고 주어진다. ($1 \leq A_i \leq N$)

셋째 줄에 N 개의 정수 $B_1, B_2, \dots, B_N$ 이 공백을 사이에 두고 주어진다. ($B_i \in \{0, 1\}$)

출력

가능한 P 중 최댓값을 출력한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2 1 2 0 1	2
4 2 4 3 1 1 1 0 0	2
10 1 3 1 2 3 1 2 3 3 1 1 1 0 0 1 1 1 1 0 1	5

노트

빈 상자는 표에서 []로 표기한다.

첫 번째 예제에서 $X=1$ 이면 상자의 상태는 $[] \rightarrow [1] \rightarrow [2]$ 가 된다. 따라서 확인한 값들은 1,2이다.

$X=2$ 이면 상자의 상태는 $[] \rightarrow [1] \rightarrow [1,2]$ 가 된다. 따라서 확인한 값들은 1,1이다.

두 경우의 결과가 다르므로 $1 \leq X \leq 2$ 라는 정보가 있으면 X 를 정확히 알아낼 수 있다.

두 번째 예제에서는 각 X 에 대해 확인한 값들의 수열이 다음과 같다.

X	확인한 값들의 수열
1	2,4,4,4
2	2,2,3,3
3	2,2,3,3
4	2,2,3,1

$X=2$ 일 때와 $X=3$ 일 때 확인한 값들의 수열이 같으므로, $1 \leq X \leq 3$ 이라는 정보가 주어져도 X 를 정확히 알아낼 수 없다.

따라서 $P=3$ 은 불가능하고, 가능한 최댓값은 2이다.

문제 I. 논문 공장

시간 제한 1 초 메모리 제한 1024 MB

아스테로모프(Asteromorph)는 AI로 과학의 발전을 가속한다는 사명 아래, 사람의 개입 없이도 창의적이면서 사실에 근거한 과학적 개념을 스스로 발견하는 시스템 Spacer를 만들고 있다. (자세한 내용은 arXiv:2508.17661 참고). Spacer의 핵심 철학은 의도적 탈맥락화(*deliberate decontextualization*)로, 정보를 키워드라는 원자 단위까지 분해한 뒤 그 사이에 아직 아무도 잊지 않은 연결에서 새로운 아이디어를 끌어낸다.

Spacer의 영감 엔진 누리는 수많은 논문에서 키워드를 모아 지식 그래프를 만들고, 그 안에서 새로운 연구 아이디어로 이어질 만한 키워드들을 찾아낸다.

어느 날, 누리를 만들던 동현이는 일렬로 놓인 키워드들의 사용 횟수를 보다가 한 가지 생각을 떠올렸다.

이 키워드들의 사용 횟수를 모두 같게 만들 수 있을까?

키워드들은 왼쪽부터 차례대로 1번부터 N 번까지 번호가 붙어 있다. $i(1 \leq i \leq N)$ 번 키워드는 처음에 A_i 번 사용되었다.

동현이는 논문을 원하는 만큼(0번 이상) 쓸 수 있다. 단, 논문 한 편은 서로 이웃한 두 키워드를 함께 다루어야 하며, 그런 논문을 쓰면 두 키워드의 사용 횟수가 각각 1씩 증가한다.

즉, 원하는 만큼 다음 연산을 할 수 있다.

정수 $i(1 \leq i < N)$ 를 하나 골라 A_i 와 A_{i+1} 을 각각 1씩 증가시킨다.

동현이는 모든 키워드의 사용 횟수를 같게 만들 수 있을까? 동현이를 도와주자.

입력

첫째 줄에 테스트 케이스의 수 $T(1 \leq T \leq 100)$ 가 주어진다.

각 테스트 케이스의 첫째 줄에 키워드의 개수 $N(1 \leq N \leq 1000)$ 이 주어진다. 둘째 줄에 각 키워드의 초기 사용 횟수 $A_1, A_2, \dots, A_N$ 이 공백을 사이에 두고 주어진다.

각 $i(1 \leq i \leq N)$ 에 대해 $0 \leq A_i \leq 1000$ 이며, 입력으로 주어지는 모든 N 의 합은 5000 이하이다. 입력으로 주어지는 모든 수는 정수이다.

출력

각 테스트 케이스마다 모든 키워드의 사용 횟수를 같게 만들 수 있으면 Yes, 아니면 No를 한 줄에 하나씩 출력한다.

Preliminaries - 예선대회

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
5	Yes
1	Yes
7	No
2	Yes
3 3	No
2	
1 2	
3	
1 0 1	
4	
1 0 1 0	

문제 J. 어려운 정렬 문제

시간 제한 1 초 메모리 제한 1024 MB

길이 N 의 정수열 A 가 주어진다. 수열은 서로 겹치지 않는 K 개의 구간으로 분할되어 있으며, 왼쪽부터 i 번째 구간의 길이는 l_i 이다. 다음 연산을 0회 이상 원하는 만큼 수행할 수 있다.

- K 개의 구간 중 하나를 고른다.
- 연산을 수행하기 직전의 해당 구간에 포함된 원소들의 최솟값을 m , 최댓값을 M 이라고 할 때, 그 구간에 속한 모든 원소를 각각 m 이상 M 이하의 임의의 정수로 바꾼다.

당신의 목표는 위 연산을 반복해 수열을 비내림차순으로 만드는 것이다.

모든 연산이 끝난 뒤, 최종 수열을 B 라 하자. 값 x 가 초기 수열 A 에서 등장한 횟수를 $c_A(x)$, 최종 수열 B 에서 등장한 횟수를 $c_B(x)$ 라 할 때, B 의 비용을 다음과 같이 정의한다.

$$\sum_x \max(0, c_B(x) - c_A(x))$$

수열을 비내림차순으로 만들 수 있는지 판별하고, 가능하다면 그러한 최종 수열의 최소 비용을 출력해 보자.

입력

첫째 줄에 정수 N 과 K 가 공백으로 구분되어 주어진다. ($1 \leq K \leq N \leq 200000$)

둘째 줄에 수열 A 의 원소 $A_1, A_2, \dots, A_N$ 이 공백으로 구분되어 주어진다. ($0 \leq A_i \leq 10^9$)

셋째 줄에 K 개의 정수 $l_1, l_2, \dots, l_K$ 가 공백으로 구분되어 주어진다. ($1 \leq l_i \leq N$; $\sum_{i=1}^K l_i = N$) 이는 수열이 왼쪽부터 차례대로 길이가 $l_1, l_2, \dots, l_K$ 인 K 개의 구간으로 나누어져 있음을 의미한다.

출력

수열을 비내림차순으로 만들 수 있다면, 그러한 최종 수열의 최소 비용을 출력한다. 그렇지 않다면, -1을 출력한다.

입출력 예시

표준 입력(stdin)

표준 출력(stdout)

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

1

위 예시에서, 수열 A 는 구간 $[1,2]$, $[3,3]$, $[4,5]$ 로 분할되어 있다.

최종 수열을 $[1,3,4,5,5]$ 로 만드는 것이 가능하다. 이때 초기 수열 $A = [3,1,4,2,5]$, 최종 수열 $B = [1,3,4,5,5]$ 에 대해, 최종 수열의 비용은 다음과 같이 계산할 수 있다.

x	1	2	3	4	5
$c_A(x)$	1	1	1	1	1
$c_B(x)$	1	0	1	1	2

최종 수열의 비용은 $0+0+0+0+1=1$ 이며, 이보다 더 작은 비용의 비내림차순 최종 수열을 만드는 것은 불가능하다.

표준 입력(stdin)

표준 출력(stdout)

```
4 4
3 1 2 4
1 1 1 1
```

-1

각 구간의 길이가 1이므로, 수열을 변경하는 것이 불가능하다. 따라서 최종 수열은 반드시 $[3,1,2,4]$ 이고, 비내림차순으로 만드는 것은 불가능하다.

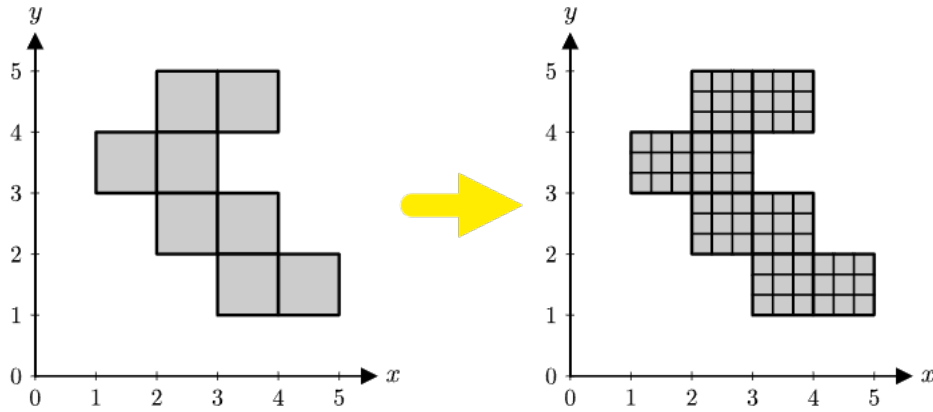
문제 K. 상인

시간 제한 5 초 메모리 제한 1024 MB

격자 위에 N 개의 마을이 있다. 각 마을은 1×1 크기의 정사각형 영역을 차지하며, i 번 마을은 좌표 (X_i, Y_i) 에 위치해 있다. 이때 좌표는 x - y 좌표계를 따른다.

각 마을을 정점으로 하고, 상하좌우로 인접한 두 마을을 간선으로 연결한 그래프를 마을의 인접 그래프라 하자. 입력으로 주어지는 마을의 인접 그래프는 트리이다. 즉, 임의의 두 마을 사이를 잇는 경로는 정확히 하나만 존재한다.

각 마을에는 9명의 상인이 살고 있다. 각 마을은 3×3 형태의 서로 같은 크기의 구역으로 나뉘며, 각 구역에는 정확히 한 명의 상인이 있다. 모든 상인은 서로 다른 종류의 물건을 판매한다. 다음은 마을과 구역의 예시이다.



마을과 구역의 예시

당신은 서로 변을 공유하는 이웃한 구역들 사이로만 이동하면서 모든 물건을 정확히 하나씩 구입하려고 한다. 어떤 구역에 방문할 때마다 그 구역의 상인에게서 반드시 물건을 하나 사야 하며, 같은 상인의 물건을 두 번 이상 구입해서는 안 된다. 또한 모든 물건을 구입한 뒤에는 출발했던 구역으로 다시 돌아와야 한다. 단, 마지막에 출발했던 구역으로 돌아왔을 때는 물건을 다시 사지 않는다.

이러한 이동이 가능한지 판별하고, 가능하다면 그 방법 중 하나를 출력하여라.

입력

첫째 줄에 테스트 케이스의 수 T 가 주어진다. ($1 \leq T \leq 10^4$)

각 테스트 케이스의 첫째 줄에는 마을의 개수 N 이 주어진다. ($1 \leq N \leq 10^5$)

이후 N 개의 줄에 걸쳐 i 번 마을의 좌표 X_i, Y_i 가 공백으로 구분되어 주어진다. ($1 \leq X_i, Y_i \leq 10^9$)

모든 마을의 좌표는 서로 다르며, 마을의 인접 그래프가 트리임이 보장된다.

모든 테스트 케이스에 대한 N 의 합은 10^5 이하이다.

출력

각 테스트 케이스마다, 조건을 만족하는 이동 방법이 존재하지 않는다면 첫째 줄에 No를 출력한다.

조건을 만족하는 이동 방법이 존재한다면 첫째 줄에 Yes를 출력한다.

이후 $9N+1$ 개의 줄에 걸쳐 이동 순서를 출력한다. 이 중 i 번째 줄에는 두 정수 a_i, b_i 를 공백으로 구분하여 출력한다. 이는 i 번째로 방문한 구역이 a_i 번 마을의 b_i 번 구역이라는 뜻이다.

이때, 각 마을에 대해 구역의 번호는 아래 그림과 같이 1부터 9까지의 정수 중 하나로 정한다.

1	2	3
4	5	6
7	8	9

구역 번호

처음 $9N$ 개 줄에 출력한 구역은 모두 서로 달라야 하며, 모든 마을의 모든 구역을 정확히 한 번씩 포함해야 한다. 즉, 처음 $9N$ 개 줄에 나타나는 순서쌍 (a_i, b_i) 는 모든 $1 \leq a_i \leq N, 1 \leq b_i \leq 9$ 에 대한 가능한 순서쌍을 각각 정확히 한 번씩 포함해야 한다.

이동 순서에서 연속한 두 구역은 변을 공유해야 한다. 또한 $(a_{9N+1}, b_{9N+1}) = (a_1, b_1)$ 이어야 한다.

가능한 이동 방법이 여러 가지라면 아무 방법이나 출력해도 좋다.

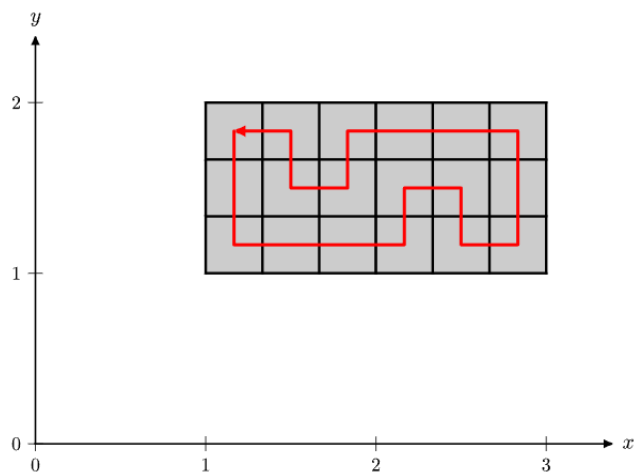
Preliminaries - 예선대회

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
2	Yes
2	1 1
1 1	1 4
2 1	1 7
1	1 8
1 1	1 9
	2 7
	2 4
	2 5
	2 8
	2 9
	2 6
	2 3
	2 2
	2 1
	1 3
	1 6
	1 5
	1 2
	1 1
	No

노트

다음은 첫 번째 테스트 케이스의 예시 그림이다. 그림은 예제 출력에서 구역을 방문하는 순서를 나타낸다.



첫 번째 테스트 케이스의 방문 순서

문제 L. 확률성 정렬

시간 제한 4 초 메모리 제한 1024 MB

길이 N 의 순열 P 가 주어진다. 배열이 오름차순이 될 때까지 다음 연산을 수행한다.

- 현재 배열에 남아 있는 원소 중 하나를 균등한 확률로 선택하여 제거한다.

남아 있는 배열이 오름차순이 될 때까지 제거한 원소의 개수의 기댓값을 구하여라.

입력

첫째 줄에 P 의 길이 N 이 주어진다. ($1 \leq N \leq 2000$)

둘째 줄에 순열의 원소 $P_1, P_2, \dots, P_N$ 이 공백으로 구분되어 주어진다. ($1 \leq P_i \leq N$)

출력

제거한 원소의 개수의 기댓값을 998244353으로 나눈 나머지를 출력하여라.

구체적으로, 가능한 모든 입력에 대해 기댓값이 항상 유리수임을 보일 수 있으며, 이를 기약분수 A/B 로 나타내었을 때 $B \times B^{-1} \equiv 1 \pmod{998244353}$ 인 정수 B^{-1} 이 항상 존재한다. 이때, $A \times B^{-1} \pmod{998244353}$ 의 값을 출력해야 한다.

입출력 예시

표준 입력(stdin)	표준 출력(stdout)
6 5 2 3 1 4 6	582309210
6 1 2 3 4 5 6	0
6 6 5 4 3 2 1	5