APC

주최 아주대학교 AJOU UNIVERSITY | SW중심대학사업 National Center of Excellence in SW 주관 ANSi 후원 Samsung Software Membership SOLVED.AC STARTLINK

Ajou Univ.

2025 아주대학교 프로그래밍 경시대회

Programming Contest



공식 문제지

Official Problemset

Division 1. Competition Round

이 문제지는 **Division 1** 참가자를 위한 문제지입니다.

- 9문제가 정상적으로 인쇄되었는지 확인하십시오.
- 모든 문제의 메모리 제한은 1024MB로 동일합니다.
- 참가자는 대회 중 사전에 승인받은 하드카피를 참고할 수 있습니다.
- 본 대회의 문제는 출제자가 판단한 난이도 순으로 정렬되어 있습니다. 다만, 실제 체감 난이도는 참가자마다 다를 수 있으므로 모든 문제를 확인해 보시기를 권장합니다.
- 대회 중 문제에 관한 질문은 대회 페이지 내 질문 기능을 이용해주시기 바랍니다.
- 대회 중 퇴실은 불가합니다.

문항 목록 - Division 1.

	문제	시간 제한
A	마지막 수강신청	1 초
B	카드 뭉치	1 초
C	Was It a Cat I Saw	1 초
D	가오리 그래프	1 초
E	공중화장실 공리주의	1 초
F	양과 늑대	3 초
G	구간 단속 종료 지점입니다.	2 초
H	2025 만들기	1 초
I	캡틴박 카페에 다녀왔습니다	1 초

문제 A. 마지막 수강신청

- 시간 제한 1초

문제

도현이는 어느덧 졸업을 앞두고 마지막 학기 수강신청을 하고 있다. 도현이는 졸업하기 전 학교에서 마지막으로 수강하고 싶은 후보 과목 N 개를 정리했다. 각 과목은 다음과 같은 정보를 가지고 있다:

- c_i: i 번째 과목을 수강하여 얻을 수 있는 학점을 나타낸다.
- s_i: 매주 i 번째 과목이 진행되는 강의 횟수를 나타낸다.
- $d_{i,j}$: 매주 i 번째 과목의 j 번째 강의가 열리는 요일을 나타낸다.
- $h_{i,j}$: 매주 i 번째 과목의 j 번째 강의가 열리는 시각을 나타낸다. 강의는 $h_{i,j}$ 시 정각에 시작해서 $h_{i,j}$ 시 59 분에 종료된다.

도현이는 이 강의들의 일부만으로 시간표를 구성했을 때, 졸업 요건인 M 학점을 수강할 수 있는지 알고 싶다. 수강하고자 하는 과목들 간 시간이 겹치지 않아야 하며, 이동시간 등은 고려하지 않는다.

입력

첫 번째 줄에 수강하고 싶은 후보 과목의 개수 N, 졸업을 위해 수강해야 하는 최소 학점 M 이 공백으로 구분되어 주어진다. $(1 \le N \le 10; 1 \le M \le 24)$

두 번째 줄부터 N 줄에 걸쳐 각각 후보 과목에 대한 정보가 주어진다. 각 줄에 c_i, s_i, $d_{i,1}, h_{i,1}, \ldots, d_{i,s_i}, h_{i,s_i}$ 가 공백으로 구분되어 주어진다. $(1 \le c_i \le 4; 1 \le s_i \le 3; 0 \le h_{i,j} \le 23;$ $d_{i,j}$ 는 `MON, TUE, WED, THU, FRI`중 하나의 값을 가진다.)

같은 과목의 강의 시간들은 오름차순으로 주어지며, 서로 다른 강의 시간이 겹치는 입력은 주어지지 않는다.

출력

후보 과목으로 시간표를 구성하여 M 학점 이상 수강할 수 있다면 `YES`, 아니라면 `NO`를 출력한다.

예제

standard input	standard output
3 6 3 2 MON 14 FRI 12 3 1 MON 14 2 2 TUE 16 FRI 12	NO
3 6 3 2 MON 14 FRI 12 2 1 MON 14 3 2 TUE 16 FRI 13	YES

노트

요일의 순서는 `MON` $<$ `TUE` $<$ `WED` $<$ `THU` $<$ `FRI` 로 정의되며, 각각 순서대로 월요일부터 금요일까지를 나타낸다.

문제 B. 카드 뭉치

- 시간 제한 1초

문제

앞면에 양의 정수가 적혀 있는 N 장의 카드 뭉치가 있다. 처음에 카드는 앞면이 위로 보이도록 일렬로 나열되어 있고, 왼쪽에서 i 번째 카드에 적혀있는 정수는 A_i 이다.

여러분은 아래의 규칙에 따라 카드 뭉치를 나눠 여러 개의 카드 뭉치로 만들어야 한다.

- 카드 뭉치를 나눌 때는 연속된 구간 단위로 분할하여 나눠야 한다.
- 카드 뭉치 내 카드의 순서는 바꿀 수 없다.
- 각 카드 뭉치에 속한 카드의 개수는 카드 뭉치의 가장 왼쪽에 있는 카드에 적힌 수보다 작거나 같아야 한다.
- 모든 카드는 정확히 하나의 카드 뭉치에 속해야 한다.

예를 들어 카드 뭉치에 4 장의 카드가 있고 왼쪽부터 카드에 $2, 3, 1, 1$ 이 적혀 있다고 하자. 위의 규칙에 따라 $[2, 3], [1], [1]$ 또는 $[2], [3, 1, 1]$ 로 카드 뭉치를 나눌 수 있지만 $[2, 1], [3, 1]$ 또는 $[2, 3, 1], [1]$ 로는 나눌 수 없다.

위의 규칙을 만족하면서 카드 뭉치의 개수가 최소가 되도록 나눴을 때, 그 개수를 구해보자.

입력

첫 번째 줄에 카드의 수 N 이 주어진다. $(1 \leq N \leq 3\,000)$

두 번째 줄에 각 카드에 적힌 양의 정수가 공백으로 구분되어 주어진다. i 번째 정수는 왼쪽에서 i 번째 카드에 적힌 정수를 의미한다. $(1 \leq A_i \leq N)$

출력

문제의 규칙을 만족하면서 카드 뭉치의 개수가 최소가 되도록 나눴을 때, 그 개수를 출력한다.

예제

standard input	standard output
4 2 3 1 1	2
5 1 2 3 4 5	3
6 6 5 4 3 2 1	1

문제 C. Was It a Cat I Saw

- 시간 제한 1초

문제



문제 제목의 문장을 거꾸로 읽어보자. 그렇다. 당신은 팰린드롬을 좋아한다.

팰린드롬(Palindrome)이란 앞으로 읽어도, 뒤로 읽어도 같은 문자열을 의미한다. 양의 정수 X가 주어질 때, 당신은 연산마다 아래의 연산 중 하나를 골라 시행할 수 있다.

- X에 1을 더한다.
- X에 1을 뺀다. 단, $X > 1$를 만족해야 한다.

X를 이진수로 표현한 문자열이 팰린드롬이 되도록 원하는 만큼 연산을 적용할 때, 필요한 연산의 최소 횟수를 구해보자. 이때 X를 이진수로 표현했을 때 앞쪽의 불필요한 0들(leading zero)은 무시한다. 예를 들어, $X = 9 = 1001_{(2)}$는 이진수로 표현했을 때 팰린드롬이지만, $X = 8 = 1000_{(2)}$는 이진수로 표현했을 때 팰린드롬이 아니다.

입력

첫 번째 줄에 테스트 케이스의 개수 T가 주어진다. $(1 \leq T \leq 30)$

두 번째 줄부터 T줄에 걸쳐 양의 정수 X가 주어진다. $(1 \leq X \leq 10^9)$

출력

각 테스트 케이스마다 X를 이진수로 표현한 문자열이 팰린드롬이 되도록 문제의 연산을 적용할 때, 필요한 연산의 최소 횟수를 출력한다.

예제

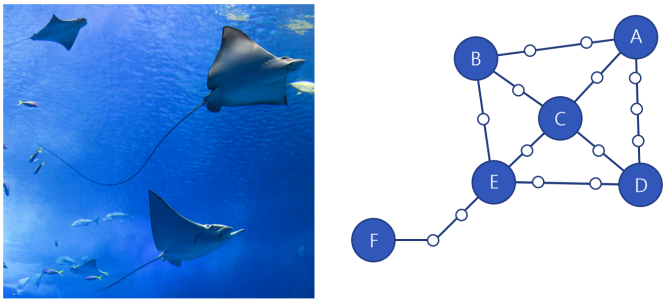
standard input	standard output
2 8 1	1 0

문제 D. 가오리 그래프

- 시간 제한 1초

문제

상혁이는 도서관에서 책을 빌려 보던 중, 책 한 장에 인상적인 가오리 그림이 그려진 페이지를 발견했다! 하지만 그림 위에 누군가 해 놓은 낙서를



왼쪽 : 가오리 그림, 오른쪽 : 가오리 그림을 따라서 그린 그래프

발견했고, 자세히 보니 단순한 낙서가 아니라 가오리 그림을 따라서 그린 그래프였다. 이 낙서에 감명을 받은 상혁이는 여기에 '가오리 그래프'라는 이름을 붙였다.

가오리 그래프는 N 개의 정점과 $N+3$ 개의 간선으로 이루어진 무방향 연결 그래프이며 아래와 같은 특징을 가지고 있다.

- 각 정점은 6 개의 핵심 정점과 $N-6$ 개의 일반 정점으로 구분된다. 핵심 정점은 머리(A), 왼쪽 날개(B), 중심(C), 오른쪽 날개(D), 아래쪽 날개(E), 꼬리(F) 이다.
- 핵심 정점 쌍 중 (A,B), (A,C), (A,D), (B,C), (B,E), (C,D), (C,E), (D,E), (E,F) 에 대해, 다른 핵심 정점을 거치지 않는 단순 경로가 유일하게 존재한다. 나머지 핵심 정점 쌍은 그러한 경로가 존재하지 않는다.
- 일반 정점은 핵심 정점 간 경로상에 위치한 정점으로, 서로 다른 2 개의 정점과 인접해 있다.

상혁이는 가오리 그래프의 각 정점에 1 번부터 N 번까지 번호를 매기고, 간선에 대한 정보를 메모했다.

시간이 흐르고, 상혁이는 우연히 자신이 작성했던 가오리 그래프에 대한 메모를 발견했다. 하지만 상혁이는 책을 반납한 뒤여서 가오리 그래프의 원래 모습을 떠올릴 수 없었다. 하지만 오른쪽 날개(D)의 번호는 왼쪽 날개(B)의 번호보다 크다는 점은 알고 있다. 상혁이를 위해 가오리 그래프의 핵심 정점이 무엇인지 알려주자.

입력

첫 번째 줄에는 정점의 수 N 이 주어진다. $(6 \le N \le 50)$

다음 줄부터 $N+3$ 개의 줄에 걸쳐 간선에 대한 정보로, 간선이 연결하는 두 정점 U, V 가 공백으로 구분되어 주어진다. $(1 \le U, V \le N; U \neq V)$

입력으로 주어지는 그래프는 가오리 그래프임이 보장된다.

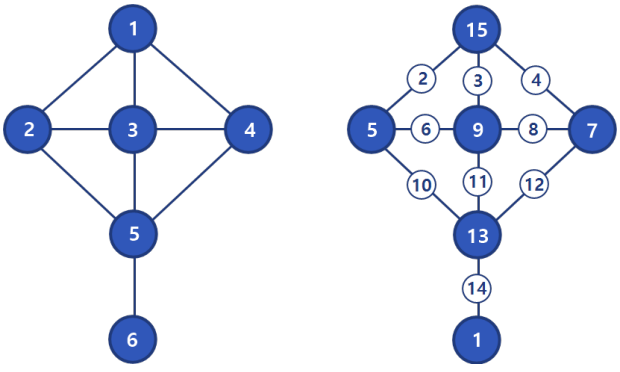
출력

첫 번째 줄에 머리, 왼쪽 날개, 중심, 오른쪽 날개, 아래쪽 날개, 꼬리를 의미하는 핵심 정점 A, B, C, D, E, F 의 번호를 공백으로 구분하여 출력한다. 입력으로 주어지는 조건에 해당하는 가오리 그래프가 유일하게 존재함을 증명할 수 있다.

예제

standard input	standard output
6 1 2 1 3 1 4 2 3 2 5 3 4 3 5 4 5 5 6	1 2 3 4 5 6
15 14 13 15 2 2 5 3 15 1 14 3 9 7 4 5 6 9 6 13 11 9 8 4 15 8 7 5 10 10 13 9 11 7 12 12 13	15 5 9 7 13 1

힌트



왼쪽 : 예제 1의 가오리 그래프, 오른쪽 : 예제 2의 가오리 그래프

문제 E. 공중화장실 공리주의

- 시간 제한 1초

문제

어떤 공중화장실에는 일렬로 나란히 놓인 소변기 N 개가 있다. 우리가 공중화장실을 이용할 때, 말로 표현되지는 않지만 많은 사람들이 공유하는 은근한 감정이 존재한다. 그것은 바로 "옆 칸에 누가 있는 상황은 가능한 피하고 싶다"는 것.

당신은 군중심리학자로서, 화장실을 이용할 때의 이러한 감정을 수치로 정량화하는 데 성공했다. 화장실을 이용하는 사람 p_i 가 느끼는 심리적 안정감 s_i 는 다음과 같이 계산된다:

- 양옆에 모두 사람이 있는 경우: $s_i = 0$
- 한쪽에만 사람이 있는 경우: $s_i = 1$
- 양쪽 모두 비어 있는 경우: $s_i = 2$

단, 양옆에 소변기가 존재하지 않는 칸은 비어 있는 것으로 간주한다.

오늘은 M 명의 사람이 동시에 화장실에 입장해 각자 소변기를 사용하려고 한다. 당신은 화장실을 이용하는 사람들의 최대 다수가 최대의 행복을 누리도록 돕고 싶기 때문에, M 명 모두가 느낄 심리적 안정감의 총합이 최대가 되게 하는 배치를 고민하고 있다. 이에 앞서, 이러한 배치의 경우의 수가 얼마나 많이 있을지 궁금해진 당신은 이를 미리 계산해 보기로 했다. 단, 동일한 소변기 집합이 배치에 사용되었다면 사람들의 순서가 바뀌더라도 같은 경우로 간주한다.

입력

첫 번째 줄에 화장실에 있는 소변기의 개수 N $(2 \leq N \leq 1\,000)$, 화장실을 이용하는 사람 수 M $(1 \leq M \leq N)$ 이 공백으로 구분되어 주어진다.

출력

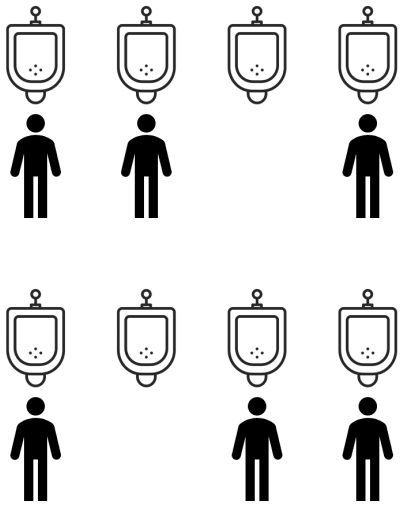
화장실을 이용하는 모든 사람들의 심리적 안정감이 총합이 최대가 되는 인원 배치의 경우의 수를 $1\,000\,000\,007$ 로 나눈 나머지를 출력한다.

예제

standard input	standard output
4 3	2
5 3	1

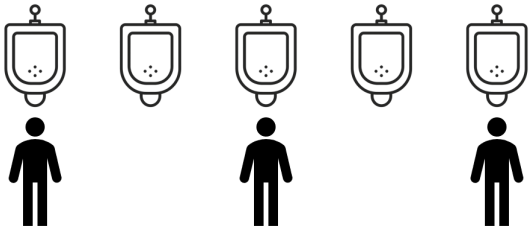
1. 아래는 첫 번째 예시에 대한 설명이다:

   예제 1의 경우, 달성 가능한 심리적 안정감의 총합의 최대는 4이고, 해당 심리적 안정감의 총합을 만족하는 배치는 아래 두 가지만 가능하다.



2. 아래는 두 번째 예시에 대한 설명이다:

   예제 2의 경우, 달성 가능한 심리적 안정감의 총합의 최대는 6이고, 해당 심리적 안정감의 총합을 만족하는 배치는 아래가 유일하다.



문제 F. 양과 늑대

- 시간 제한 3초

문제

양치기 소년은 외딴 목장에서 N 마리의 양과 M 마리의 늑대를 기르고 있다.

각 양은 1 번부터 N 번까지 번호가 붙어 있으며 i 번 양의 크기는 A_i 이다. 각 늑대 또한 1 번부터 M 번까지 번호가 붙어 있으며 j 번 늑대의 크기는 B_j 이다.

양치기 소년은 자신이 기르고 있는 동물들을 관리하기 위해 우리를 만들고자 한다. 양치기 소년이 만든 우리는 최대 2 마리의 동물이 들어갈 수 있으며, 우리에 들어간 동물의 크기의 합은 K 를 초과할 수 없다. 각 우리에는 같은 종류의 동물만 들어갈 수 있지만, 예외적으로 양치기 소년이 감시할 수 있는 우리 하나에는 양과 늑대가 같이 들어갈 수 있다. 또한 모든 동물의 크기는 K 를 넘지 않는다.

모든 동물을 우리에 넣어야 할 때, 양치기 소년이 만들어야 하는 우리는 최소 몇 개인지 구해보자.

입력

첫 번째 줄에 N, M, K 가 공백으로 구분되어 주어진다. $(1 \le N, M \le 500\,000; 1 \le K \le 10^9)$

두 번째 줄에 양의 크기를 의미하는 $A_1, A_2, \cdots, A_N$ 가 공백으로 구분되어 주어진다. $(1 \le A_i \le K)$

세 번째 줄에 늑대의 크기를 의미하는 $B_1, B_2, \cdots, B_M$ 가 공백으로 구분되어 주어진다. $(1 \le B_j \le K)$

입력으로 주어지는 모든 값은 정수이다.

출력

첫 번째 줄에 양치기 소년이 만들어야 하는 우리의 최소 개수를 출력한다.

예제

standard input	standard output
4 4 8 3 7 4 2 7 4 7 8	6
5 4 2 1 1 1 1 1 1 1 1 1	5
1 1 5 2 3	1

문제 G. 구간 단속 종료 지점입니다.

- 시간 제한 2초

문제

2025 아주대학교 프로그래밍 경시대회가 끝난 후, 현빈이는 차를 타고 본가에 내려가는 중이다.

현빈이가 본가로 가는 경로는 일직선으로 나타낼 수 있으며, 학교는 $x = 0$에, 본가는 $x = E$에 위치하고 있다. 현빈이가 운전하는 차의 속력은 M을 넘을 수 없다.

본가로 향하는 길에는 총 N개의 구간 단속 지점이 있다. $i(1 \le i \le N)$번째 구간 단속 지점 $[s_i, e_i)$는 $x = s_i$에서 시작하고 $x = e_i$에서 종료되며, 구간을 지나는 평균 속도는 v_i를 초과해서는 안 된다.

학교의 위치인 $x = 0$에서 출발하여 현빈이의 본가인 $x = E$에 도착하는데 걸린 최소 시간을 구해보자.

단, 현빈이의 차가 가속과 감속을 하는 데는 시간이 걸리지 않는다.

입력

첫 번째 줄에 구간 단속 지점의 수 N, 현빈이가 운전하는 차의 최대 속력 M, 현빈이의 본가 위치 E가 공백으로 구분되어 주어진다. $(1 \le N \le 500\,000;$ $1 \le M \le 1\,235; 1 \le E \le 10^9)$

두 번째 줄부터 N개 줄에 걸쳐 구간 단속 지점에 대한 정보 s_i, e_i, v_i가 공백으로 구분되어 주어진다. $(0 \le s_i < e_i \le E; 1 \le v_i \le M)$

입력으로 주어지는 모든 수는 정수이다.

출력

현빈이가 본가에 도착하는데 걸린 최소 시간을 출력한다. 절대/상대 오차는 10^{-6}까지 허용한다.

예제

standard input	standard output
2 5 20 5 10 2 10 15 2	7
2 5 20 5 10 2 7 15 1	10.4

노트

- $x = s$를 통과하는 시점이 t_1이고 $x = e$를 통과하는 시점이 t_2라고 할 때, 구간 $[s, e)$의 평균 속도 $\bar{v}$는 다음과 같다.

$$\bar{v} = \frac{e - s}{t_2 - t_1}$$

- 현빈이의 집은 용인시 수지구로, 고작 차로 10분 거리라고 한다.

문제 H. 2025 만들기

- 시간 제한 1초

문제

1 부터 N 까지의 N 개의 정수로 이루어진 배열 A 가 주어질 때 다음 3 가지 연산 중 원하는 연산을 골라 시행하는 것을 $N-1$ 번 진행한다.

1. 배열의 두 원소 a, b 를 제거하고 $a+b$ 를 삽입한다.
2. 배열의 두 원소 a, b 를 제거하고 $a-b$ 를 삽입한다.
3. 배열의 두 원소 a, b 를 제거하고 $a \times b$ 를 삽입한다.

연산을 $N-1$ 번 시행한 뒤 마지막으로 남은 수가 2025 일 수 있는지와 가능한 경우 구성 방법까지 구해보자. 단, 계산 과정에서 계산 결과의 절댓값은 10^9 를 넘으면 안 된다.

입력

배열의 길이 N 이 주어진다. $(1 \leq N \leq 100\,000)$

출력

연산을 $N-1$ 번 시행한 뒤 마지막으로 남은 수가 2025 일 수 있다면, 첫째 줄에 YES를 출력하고, 불가능하다면 NO를 출력한다.

마지막으로 남은 수가 2025 일 수 있는 경우, 이후 $N-1$ 개의 줄에 걸쳐 마지막으로 남은 수가 2025 가 되도록 시행할 연산을 순서대로 출력한다. 출력 형식은 구체적으로 다음과 같다.

- 각 줄에는 "`<num1> <op> <num2>`" 의 형식으로 시행한 연산의 정보를 출력한다.
- `<num1>`과 `<num2>`는 연산을 시행하기 이전 배열에 남아있는 서로 다른 두 원소이다.
- `<op>`는 `+`, `-`, `*`의 세 가지 문자 중 하나이다. 각각 1번, 2번, 3번 연산의 기호를 의미한다.
- `<num1>`과 `<num2>`가 음이 아닌 정수인 경우, 정수 앞에 부호를 붙여서 출력하면 안 된다.
- 연산을 수행했을 때 나온 계산 결과의 절댓값은 10^9 이하여야 한다.

예제

standard input	standard output
9	YES 1 + 2 3 + 4 5 + 6 7 + 8 9 * 15 3 - 7 11 - -4 135 * 15
2	NO

문제 I. 캡틴박 카페 다녀왔습니다

- 시간 제한 1초

문제

> "어제 박지성 카페 다녀왔습니다. 박지성 카페가 열린 건 아니고요, 그냥 카페에서 박지성 생각했습니다. 카페에 간 건 아니고요, 그냥 집에서 커피를 마셨습니다. 사실 커피도 안마셨습니다. 그냥 캡틴박인 상태입니다"
>
> – 원문 : 대흥민 카페 다녀왔습니다

박지성의 열렬한 팬인 종학이는 종종 자신이 박지성이라고 상상하곤 한다. 대한민국의 캡틴 박지성은, 언제나 그렇듯 자신을 향해 달려오는 수비수들을 피해 최대한 오래 드리블을 해야 한다.

어이없겠지만 축구장을 N 개의 정점과 서로 다른 두 정점을 잇는 $N-1$ 개의 양방향 간선으로 구성된 트리라고 생각해 보자. 정점은 1 번부터 시작하며, i 번째 간선은 정점 a_i 와 b_i 를 잇는 길이가 w_i 인 직선 구간이다. 이때, 모든 간선의 길이 w_i 는 짝수로만 주어진다. $(1 \leq i \leq N-1)$ 경기장에는 지점이 존재하며, 이는 정점 또는 간선 위에서 정점으로부터의 거리가 정수인 위치를 의미한다.

경기가 시작할 때 박지성은 1 번 정점에, 수비수들은 1 번 정점을 제외한 K 개의 정점에 위치한다. 박지성은 수비수들을 피해 최대한 긴 시간 동안 드리블을 하고자 한다.

매초가 시작되면, 박지성과 K 명의 수비수들은 순서에 따라 다음과 같이 행동한다:

1. 박지성은 현재 정점에 머무르거나, 수비수가 없는 인접한 지점으로 간선을 따라 1 만큼 뛰어간다.
2. 각 수비수는 박지성이 위치한 지점으로의 최단 경로 방향으로 간선을 따라 1 만큼 뛰어간다.

이동 후, 박지성과 수비수들은 같은 지점에 동시에 여러 명이 위치할 수 있다. 또한 박지성이 현재 정점에 머무르고 있는 상황 역시 드리블로 간주한다.

만약 박지성과 어떤 수비수가 같은 지점에 위치하게 된다면, 박지성은 아쉽게도 그 수비수에게 저지당하게 된다. 박지성이 수비수들을 피해 드리블할 수 있는 최대 시간을 출력해 보자.

모든 입력에 대해 정답은 항상 정수임을 보일 수 있다.

입력

첫 번째 줄에 정점의 개수 N, 수비수의 명수 K 가 공백으로 구분되어 주어진다.$(2 \leq N \leq 200\,000; 1 \leq K \leq N-1)$

두 번째 줄부터 $N-1$ 개의 줄에 걸쳐 세 개의 정수 a_i, b_i, w_i 가 공백으로 구분되어 주어진다. 이는 정점 a_i 와 b_i 를 연결하는 길이가 w_i 인 간선이 존재한다는 의미이다. $(1 \leq i \leq N-1; 1 \leq a_i, b_i \leq N; a_i \neq b_i; 1 \leq w_i \leq 10^9; w_i \equiv 0 \pmod 2)$

마지막 줄에는 K 개의 정수 p_j 가 공백으로 구분되어 주어지며, 이는 초기에 j 번째 수비수가 정점 p_j 에 위치해 있다는 의미이다. 입력으로 주어지는 수비수의 초기 위치는 서로 다르다. $(1 \leq j \leq K; 2 \leq p_j \leq N)$

출력

박지성이 수비수에게 저지당하지 않고 드리블할 수 있는 최대 시간을 출력한다.

예제

standard input	standard output
4 1 1 2 4 2 3 4 2 4 6 3	8
3 2 1 2 2 1 3 4 2 3	3
6 2 1 2 4 1 6 4 2 3 4 3 4 8 2 5 2 5 6	5

마지막 페이지입니다.