SNUPC 2026

Division 1 문제지

2026년 9월 13일 일요일 13:00 - 17:00 (KST)

문제 목록

이 문제지는 A부터 H까지 총 8문제이며, 표지를 포함하여 총 10쪽입니다.
본인의 Division과 문제 번호를 확인해 주세요.

번호	문제	시간	메모리	쪽
A	MEX Game	1초	1024 MiB	2
B	MOVING MST	3초	1024 MiB	3
C	반전 술식	1초	1024 MiB	4
D	드래곤 슬레이어	1초	1024 MiB	5
E	외계 선인장 리턴즈	1초	1024 MiB	6
F	Yet Another Permutation Problem	1초	1024 MiB	7
G	Disjoint Paths	1초	1024 MiB	8
H	관악 마운틴 노루 점핑	2초	1024 MiB	9

모든 문제의 입력과 출력은 표준 입력과 표준 출력을 사용합니다.

문제 A. MEX Game

시간 제한	1초
메모리 제한	1024 MiB

테라와 루루는 정수가 적힌 카드 N장을 가지고 게임을 한다.

처음에는 모든 카드가 바닥에 놓여 있다. 테라가 먼저 시작하며, 두 사람은 번갈아 턴을 진행한다.

자신의 턴이 되면 바닥에 남아 있는 카드 중 한 장 이상을 원하는 만큼 골라 가져간다. 이번 턴에 가져간 카드에 적힌 정수들의 멀티셋을 S라고 하자. 현재 플레이어는 이번 턴에 다음 점수를 얻는다.

$$\mathrm{mex}(S) - |S|$$

여기서 $\mathrm{mex}(S)$는 S에 포함되지 않은 가장 작은 음이 아닌 정수이고, $|S|$는 이번 턴에 가져간 카드의 수이다.

바닥에 카드가 한 장도 남아 있지 않으면 게임이 종료된다. 각 플레이어의 최종 점수는 자신의 모든 턴에서 얻은 점수의 합이다.

두 사람은 자신의 최종 점수에서 상대의 최종 점수를 뺀 값을 최대화하도록 최적으로 플레이한다.

테라의 최종 점수가 더 크다면 `Terra`, 루루의 최종 점수가 더 크다면 `Lulu`, 두 점수가 같다면 `Sphere`를 출력하라.

입력

첫째 줄에 테스트 케이스의 개수 T가 주어진다. $(1 \le T \le 10\,000)$

각 테스트 케이스는 다음과 같이 주어진다.

첫째 줄에 카드의 수 N이 주어진다. $(1 \le N \le 200\,000)$

둘째 줄에 각 카드에 적힌 정수 $A_1, A_2, \ldots, A_N$이 공백으로 구분되어 주어진다. $(0 \le A_i \le N)$

모든 테스트 케이스에 대한 N의 합은 $200\,000$ 이하이다.

출력

각 테스트 케이스마다 테라가 이기면 `Terra`, 루루가 이기면 `Lulu`, 비기면 `Sphere`를 출력한다.

입출력 예시

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

문제 B. MOVING MST

시간 제한	3초
메모리 제한	1024 MiB

정점 N개와 간선 M개로 이루어진 연결된 무방향 가중치 그래프가 주어진다. 정점에는 1부터 N까지 번호가 매겨져 있다. 그래프에는 중복 간선이 주어질 수 **있다**.

당신은 그래프에 다음 연산을 수행할 수 있다.

- 그래프에 존재하는 간선 하나를 선택한다.
- 선택한 간선의 가중치는 유지한 채, 간선을 원하는 두 정점 사이로 옮긴다.

연산을 수행한 결과 두 정점 사이에 여러 개의 간선이 존재할 수 **있다**.

각 $K = 1, 2, \cdots, N$에 대하여, 위 연산을 최대 K번 수행하여 만들 수 있는 그래프의 최소 스패닝 트리 가중치 합의 최솟값을 구하여라.

입력

첫째 줄에 두 정수 N, M이 공백으로 구분되어 주어진다. $(1 \le N \le 200\,000;\ N - 1 \le M \le 200\,000)$

둘째 줄부터 M개의 줄에 걸쳐 세 정수 u_i, v_i, w_i가 공백으로 구분되어 주어진다.

$(1 \le u_i, v_i \le N;\ 1 \le w_i \le 10^9;\ u_i \ne v_i)$

이는 정점 u_i와 정점 v_i 사이에 가중치가 w_i인 무방향 간선이 있음을 의미한다.

주어지는 그래프는 연결 그래프이다.

출력

N개의 줄에 걸쳐 i번째 줄에는 $K = i$일 때의 답을 출력한다.

입출력 예시

표준 입력 (stdin)	표준 출력 (stdout)
4 5 1 2 1 1 2 2 1 2 3 2 3 100 3 4 200	103 6 6 6

문제 C. 반전 술식

시간 제한	1초
메모리 제한	1024 MiB

정점 N개로 이루어진 트리 T가 있다. 트리의 모든 간선은 유향 간선이다.
정점 u에 대해 **좋은 트리**란, 트리의 모든 정점에서 유향 간선들만을 따라 정점 u에 도달할 수 있는 트리를 말한다.
유향 트리 T'를 정점 u에 대해 좋은 트리로 만들기 위해 다음 연산을 수행할 수 있다.

- 트리에서 서로 다른 두 정점을 잇는 단순 경로 하나를 고른 후, 해당 경로를 이루는 모든 간선의 방향을 뒤집는다.

T'를 정점 u에 대해 좋은 트리로 만들기 위해 필요한 최소 연산 횟수를 $f_u(T')$라 하자.
당신은 다음 Q개의 쿼리를 처리해야 한다.

- `1 x y`: 트리 T에서 정점 x와 정점 y를 잇는 간선의 방향을 뒤집는다. 해당 간선이 존재함이 보장된다.
- `2 u`: $f_u(T)$를 구한다.

입력

첫째 줄에 두 정수 N, Q가 공백으로 구분되어 주어진다. $(1 \le N, Q \le 500\,000)$
둘째 줄부터 $N-1$개의 줄에 걸쳐 두 정수 u, v가 공백으로 구분되어 주어진다. 이는 u에서 v로 향하는 유향 간선 $u \to v$가 존재함을 의미한다. $(1 \le u, v \le N)$
그다음 줄부터 Q개의 줄에 걸쳐 쿼리가 주어진다. 2번 쿼리는 적어도 1개 이상 주어진다.

출력

모든 2번 쿼리에 대해 답을 한 줄에 하나씩 출력한다.

입출력 예시

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

문제 D. 드래곤 슬레이어

시간 제한	1초
메모리 제한	1024 MiB

각각 머리가 $H_1, H_2, \cdots, H_N$개 달린 용이 N마리 살고 있다. 당신은 원하는 용을 지목하고, 다음 Q개의 공격 중 하나를 사용할 수 있다. 공격은 원하는 만큼 사용할 수 있으며, 서로 다른 공격을 사용하여도 된다. 단, 공격 도중에 머리가 0개 미만이 되면 안 된다.

- a_j b_j: 지목한 용의 머리가 a_j개 이상이라면 a_j개의 머리를 잘라 없앤다. 이후 해당 용에게 b_j개의 머리가 자라난다.

공격이 모두 끝난 후 머리가 정확히 0개인 용을 **완벽히 처치된 용**이라고 부른다. 각 $i = 1, 2, \cdots, N$에 대하여, i번 용이 완벽히 처치된 용이 될 수 있는지 판별하여라.

입력

첫째 줄에 두 정수 N, Q가 공백으로 구분되어 주어진다. $(1 \le N, Q \le 200\,000)$

둘째 줄에 N개의 양의 정수 $H_1, H_2, \cdots, H_N$이 공백으로 구분되어 주어진다. $(1 \le H_i \le 10^{18})$

셋째 줄부터 Q개의 줄에 걸쳐 두 정수 a_j, b_j가 공백으로 구분되어 주어진다. $(0 \le a_j, b_j \le 500)$

출력

N개의 줄에 걸쳐 i번째 줄에 i번 용이 완벽히 처치된 용이 될 수 있다면 `YES`를, 그렇지 않다면 `NO`를 출력한다.

입출력 예시

표준 입력 (stdin)	표준 출력 (stdout)
5 2 1 2 3 5 10 2 0 4 1	NO YES NO YES YES

문제 E. 외계 선인장 리턴즈

시간 제한	1초
메모리 제한	1024 MiB

어떤 외계 행성의 지표면은 사막이 대부분이라 비가 별로 오지 않는다. 이 행성에 사는 식물은 모두 선인장이다. 비가 워낙 오지 않기 때문에, 선인장들은 서로 협력해서 물을 저장하는 방법을 가지도록 진화했다.

행성의 한 지역은 상하 방향과 좌우 방향만 존재하는 2차원 평면이다. 그 지역에 N개의 선인장이 일렬로 붙어 서 있다. 선인장의 폭은 모두 1미터이며, i번째 선인장의 높이는 H_i미터 이다.

환경미화를 위해 l번째 선인장에서 r번째 선인장까지만 남긴 상태에서 비가 충분히 많이 내렸다. 범위 밖의 선인장은 모두 없어진 것으로 생각한다.

당신은 비가 오기 전에 a번째 선인장부터 b번째 선인장까지 비료를 뿌릴 수 있다. 비료를 뿌린 선인장의 높이는 모두 k만큼 증가한다. 비가 내린 뒤 선인장 사이에서 물이 흘러나가지 않고 담길 수 있는 모든 영역에 물이 고인다. 고인 물의 양은 물이 차지하는 영역의 넓이이다.

선인장들의 높이와 남아 있는 선인장들의 범위가 주어질 때, 적절한 a, b, k를 정하여 고인 물의 양을 최대화하여라.

남아 있는 선인장들의 범위는 Q번 주어진다. 각 쿼리는 비료를 뿌리지 않은 초기 상태에서 독립적으로 수행된다.

입력

첫째 줄에 두 정수 N, Q가 공백으로 구분되어 주어진다. $(1 \le N, Q \le 200\,000)$

둘째 줄에 N개의 양의 정수 $H_1, H_2, \cdots, H_N$이 공백으로 구분되어 주어진다. $(1 \le H_i \le 10^9)$

셋째 줄부터 Q개의 줄에 걸쳐 두 정수 l, r이 공백으로 구분되어 주어진다. $(1 \le l \le r \le N)$

출력

각 쿼리마다 한 줄에 네 정수 $a, b, k,$ answer를 공백으로 구분하여 출력한다. a, b는 비료를 뿌릴 범위이며, k는 선인장의 성장 정도이다. answer는 해당 비료를 뿌린 뒤 고일 수 있는 물의 양의 최댓값이어야 한다. $(l \le a \le b \le r,\ 0 \le k \le 10^9)$

조건을 만족하는 답이 여러 가지라면 그중 아무거나 출력한다.

입출력 예시

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

문제 F. Yet Another Permutation Problem

시간 제한	1초
메모리 제한	1024 MiB

$(1, 2, \cdots, N)$의 순열 $A = (A_1, A_2, \cdots, A_N)$이 주어진다.

처음에 현재 순열은 A이다. 현재 순열에 다음 연산을 원하는 만큼 수행할 수 있다.

현재 순열을 $(B_1, B_2, \cdots, B_N)$이라 하자. 인덱스 $i (1 \le i \le N)$를 하나 고른 뒤, $i < j \le N$이고 $B_j < B_i$인 모든 원소 B_j를 B_i의 바로 앞으로 옮긴다. 옮겨지는 원소들의 상대적인 순서는 유지되며, 그 외 원소들의 상대적인 순서도 유지된다.

예를 들어, 현재 순열이 $(1, 5, 2, 6, 4, 3)$일 때 $i = 2$를 고르면 $(1, 2, 4, 3, 5, 6)$이 된다.

연산을 0번 이상 수행하여 얻을 수 있는 서로 다른 순열의 개수를 $998\,244\,353$으로 나눈 나머지를 구하여라.

입력

첫째 줄에 정수 N이 주어진다. $(2 \le N \le 200\,000)$

둘째 줄에 N개의 정수 $A_1, A_2, \cdots, A_N$이 공백으로 구분되어 주어진다. A는 $(1, 2, \cdots, N)$의 순열이다.

출력

첫째 줄에 얻을 수 있는 서로 다른 순열의 개수를 $998\,244\,353$으로 나눈 나머지를 출력하여라.

입출력 예시

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

문제 G. Disjoint Paths

시간 제한	1초
메모리 제한	1024 MiB

$N \times M$ 격자판이 있다. 위에서부터 $1, 2, \cdots, N$ 행, 왼쪽에서부터 $1, 2, \cdots, M$ 열이라고 하며, r 행 c 열의 점을 (r, c)로 나타낸다.

오른쪽 또는 아래쪽으로만 이동하여 1 행의 점에서 시작해 N 행의 점에 도착하는 경로를 **좋은 경로**라고 하자. 서로 겹치지 않는 L 개의 좋은 경로가 주어진다. 두 경로가 겹치지 않는다는 것은 두 경로가 공통으로 지나는 점이 없다는 뜻이다.

각 경로는 시작하는 열과 이동 방법으로 주어진다. 문자 `D`는 한 칸 아래로 이동하는 것을, 문자 `R`은 한 칸 오른쪽으로 이동하는 것을 나타낸다.

좋은 경로에 다음 연산을 수행할 수 있다. 경로가 어떤 1×1 정사각형의 서로 이웃한 두 변을 따라 지나고 있다면, 그 부분을 같은 정사각형의 나머지 두 변을 따라 지나도록 바꾼다. 즉, 아래로 한 칸 이동한 뒤 오른쪽으로 한 칸 이동하는 부분과 오른쪽으로 한 칸 이동한 뒤 아래로 한 칸 이동하는 부분을 서로 바꿀 수 있다.

원하는 경로에 연산을 원하는 만큼 수행할 수 있다. 단, 연산을 수행하는 모든 순간에 모든 경로는 좋은 경로여야 하며 서로 겹치지 않아야 한다.

또한 K 개의 점이 주어진다. 각 $1 \le i \le L$ 에 대해, 처음 상태에서 시작하여 i 번째 경로가 주어진 K 개의 점을 모두 지나도록 만들려고 한다. 이 과정에서 다른 경로에도 연산을 수행해도 된다.

필요한 전체 연산 횟수의 최솟값을 C_i 라고 하자. 불가능하다면 $C_i = -1$ 로 정한다. 각 C_i 는 같은 초기 상태에서 독립적으로 구한다.

$C_1, C_2, \cdots, C_L$ 을 구하여라.

입력

첫째 줄에 네 정수 N, M, L, K 가 공백으로 구분되어 주어진다. $(2 \le N, M \le 2\,000;\ 1 \le L \le M;\ 1 \le K \le 4\,000)$

다음 L 개의 줄 중 i 번째 줄에 정수 s_i 와 `D`, `R`로만 이루어진 문자열 T_i 가 공백으로 구분되어 주어진다. $(1 \le s_i \le M)$

i 번째 경로는 $(1, s_i)$ 에서 시작한다. T_i 를 앞에서부터 읽으면서 `D`가 나오면 한 칸 아래로, `R`이 나오면 한 칸 오른쪽으로 이동한다. 각 T_i 에는 `D`가 정확히 $N - 1$ 개 존재한다. 주어지는 모든 경로는 격자판을 벗어나지 않으며, L 개의 경로는 서로 겹치지 않는다.

다음 K 개의 줄 중 j 번째 줄에 두 정수 u_j, v_j 가 공백으로 구분되어 주어진다. $(1 \le u_j \le N;\ 1 \le v_j \le M)$

주어지는 K 개의 점은 서로 다르다.

출력

첫째 줄에 $C_1, C_2, \cdots, C_L$ 을 공백으로 구분하여 출력하여라.

입출력 예시

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

문제 H. 관악 마운틴 노루 점핑

시간 제한	2초
메모리 제한	1024 MiB

관악산은 N개의 지점과 M개의 길로 이루어진 연결 그래프로 나타낼 수 있다. 각 길은 서로 다른 두 지점을 연결하며, 흙길 또는 돌길 중 하나이다.

두 개 이상의 길과 연결된 모든 지점에는 흙길과 돌길이 모두 하나 이상 연결되어 있다.

관악산에는 노루 한 마리가 살고 있다. 노루는 한 번의 이동으로 최대 3개의 길을 연속해서 지나 다른 지점으로 이동할 수 있다. 단, 정확히 3개의 길을 지나는 경우 세 길이 모두 같은 종류여서는 안 된다. 이동하는 도중 거치는 지점은 방문한 것으로 간주하지 않는다.

노루는 어떤 지점에서 출발하여 모든 지점을 정확히 한 번씩 방문한 뒤 다시 출발점으로 돌아오려고 한다.

조건을 만족하는 방문 순서를 하나 구해보자. 이러한 방문 순서가 항상 존재함을 보일 수 있다.

입력

첫째 줄에 지점의 수 N과 길의 수 M이 공백으로 구분되어 주어진다. $(1 \le N, M \le 5\,000)$

다음 M개의 줄에 길의 정보가 주어진다. 각 줄에는 세 정수 u, v, t가 공백으로 구분되어 주어진다. 이는 u번 지점과 v번 지점을 연결하는 길이 있음을 의미한다. $(1 \le u, v \le 5000; u \ne v; t \in \{0, 1\})$

$t = 0$이면 해당 길은 흙길이고, $t = 1$이면 해당 길은 돌길이다.

주어지는 그래프는 연결되어 있으며, 두 개 이상의 길과 연결된 모든 지점에는 흙길과 돌길이 모두 하나 이상 연결되어 있다. 또한 그래프에는 두 지점을 연결하는 길이 여러 개 존재하지 않는다.

출력

첫째 줄에 노루가 출발하는 지점을 출력한다.

다음 N개의 줄에는 노루의 이동 경로를 순서대로 출력한다. i번째 줄에는 i번째 이동에서 방문하는 지점의 개수 K_i와, 해당 경로에 포함된 K_i개의 지점을 순서대로 출력한다. 이때 출발 지점은 이동 경로에 포함하지 않으며 각 이동 경로의 마지막 지점은 해당 이동의 도착 지점이다.

입출력 예시

표준 입력 (stdin)	표준 출력 (stdout)
5 4 1 2 0 1 3 1 1 4 0 1 5 1	1 1 2 2 1 3 2 1 4 2 1 5 1 1
8 11 3 7 0 5 7 1 1 5 0 3 5 1 3 8 0 2 8 1 1 2 0 1 6 1 4 6 0 2 4 1 2 3 0	3 1 7 3 3 8 2 3 1 6 4 1 6 3 1 2 8 3 3 7 5 1 1 3 5 7 3