SNUPC 2026

Division 2 문제지

2026년 9월 13일 일요일 13:30 - 16:30 (KST)

문제 목록

이 문제지는 A부터 H까지 총 8문제이며, 표지를 포함하여 총 9쪽입니다.
본인의 Division과 문제 번호를 확인해 주세요.

번호	문제	시간	메모리	쪽
A	샤진 찍기	1초	256 MiB	2
B	minimax 게임	1초	1024 MiB	3
C	루루 격자	1초	1024 MiB	4
D	MEX Game	1초	1024 MiB	5
E	MOVING MST	3초	1024 MiB	6
F	반전 술식	1초	1024 MiB	7
G	드래곤 슬레이어	1초	1024 MiB	8
H	외계 선인장 리턴즈	1초	1024 MiB	9

모든 문제의 입력과 출력은 표준 입력과 표준 출력을 사용합니다.

문제 A. 샤진 찍기

시간 제한	1초
메모리 제한	256 MiB

서울대 정문이 나와있는 사진을 "샤진"이라고 한다. N명의 학생들은 서울대 정문의 기운을 받기 위해 샤진을 찍으려고 한다. 이를 위해서 각 학생들은 K일중 하루에 서울대에 방문해야 한다.

그러나 각 학생들은 샤진을 너무 늦게 찍으면 서울대 정문의 기운이 약해진다고 생각한다. 구체적으로 i번째 학생은 자신이 서울대에 방문했을 때, 그 날 이미 샤진을 찍은 학생이 A_i명 이하이기를 원한다.

서울대학교에서는 N명의 학생들이 K일중 언제 몇 번째 순서로 샤진을 찍을지 결정할 수 있다. 또한 N명의 학생들의 요구를 모두 만족하려 하며, 이를 위해 K일동안 N명의 학생들을 제외한 사람들은 모두 샤진을 찍을 수 없도록 제한하려고 한다. 하지만 N명의 학생들 이외에도 서울대에 방문할 사람이 많기 때문에, 불편을 줄이기 위해서 샤진을 찍는 기간을 최대한 짧게 하려고 한다.

SNUPS 부원들은 학생들의 방문 일정을 조정하는 역할을 맡았다. 부원들을 도와 학생들의 요구를 모두 만족할 수 있는 K의 최솟값을 찾아주자!

입력

첫째 줄에 N이 주어진다. $(1 \le N \le 100\,000)$

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

출력

첫째 줄에 K의 최솟값을 출력한다.

입출력 예시

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

문제 B. minimax 게임

시간 제한	1초
메모리 제한	1024 MiB

모그와 루날리티는 양의 정수로 이루어진 수열 $A = [A_1, A_2, \cdots, A_N]$와 정수 K를 가지고 게임을 하고 있다. 모그는 종이 하나를 가지고 있으며, 초기에 그 종이에는 A_1이 쓰여 있다.

이후 모그와 루날리티는 다음과 같은 행동을 $i = 2, 3, \cdots, N$에 대해 각각 순차적으로 수행한다.

1. 루날리티는 모그에게 양의 정수 A_i를 준다.
2. 이후 모그는 다음의 행동 중 하나를 한다.
   - 모그의 종이에 적혀 있는 수를 C라 할 때, C를 $\min(C, A_i)$로 바꾼다. 이 연산을 **최소화 연산**이라고 한다.
   - 모그의 종이에 적혀 있는 수를 C라 할 때, C를 $\max(C, A_i)$로 바꾼다. 이 연산을 **최대화 연산**이라고 한다.

둘의 목표는 모든 행동을 수행하고 난 후에 종이에 적혀있는 수를 최대화 시키는 것이다. 하지만, 둘을 방해하려고 하는 소로휴는 최대화 연산은 정확히 K번만 사용할 수 있다는 제약 조건을 주었다. 이때 마지막에 종이에 적혀있는 수의 최댓값을 구하여라.

입력

첫째 줄에 정수 N, K가 공백으로 구분되어 주어진다. $(2 \le N \le 200\,000;\ 0 \le K \le N - 1)$

둘째 줄에 양의 정수 $A_1, A_2, \cdots, A_N$이 공백으로 구분되어 주어진다. $(1 \le A_i \le 10^9)$

출력

첫째 줄에 마지막에 종이에 적혀있는 수의 최댓값을 출력한다.

입출력 예시

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

문제 C. 루루 격자

시간 제한	1초
메모리 제한	1024 MiB

평면 위에 N개의 행과 M개의 열로 이루어진 격자점이 있다. 위에서 i번째 행, 왼쪽에서 j번째 열에 있는 격자점의 좌표는 (i, j)이며, 가로 또는 세로로 인접한 두 격자점 사이의 거리는 1이다.

당신은 모든 격자점을 몇 가지 색으로 칠하려고 한다. 사용하는 색의 수를 C라 하며, 색은 1부터 C까지의 정수로 나타낸다.

이때 색칠된 격자는 다음 조건을 모두 만족해야 한다:

- 모든 격자점은 정확히 하나의 색으로 칠해져야 한다.
- 각 색은 정확히 3개의 격자점에 사용되어야 한다.
- 같은 색으로 칠해진 세 격자점은 한 직선 위에 있지 않아야 한다.
- 같은 색으로 칠해진 세 격자점을 꼭짓점으로 하는 삼각형의 넓이는 모든 색에 대해 서로 같아야 한다.

조건에 만족하는 색칠 방법이 존재하는지 판별하고, 존재한다면 색칠 방법을 하나 구하여라.

입력

첫째 줄에 두 정수 N, M이 공백으로 구분되어 주어진다. $(2 \le N, M \le 2\,000)$

출력

조건을 만족하는 색칠 방법이 존재하지 않는다면 첫째 줄에 -1을 출력한다.

조건을 만족하는 색칠 방법이 존재한다면 N개의 줄에 걸쳐 색칠 결과를 출력한다.

각 줄에는 M개의 정수를 공백으로 구분하여 출력한다. i번째 줄의 j번째 정수는 위에서 i번째 행, 왼쪽에서 j번째 열에 있는 격자점에 칠한 색을 의미한다.

조건을 만족하는 답이 여러 가지라면 그중 아무거나 출력한다.

입출력 예시

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

문제 D. 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 E. 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 F. 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 G. 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 H. 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