

Задача А. Хорошие подотрезки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана перестановка p чисел $1, \dots, n$. Назовём подотрезок $[l, r]$ хорошим, если на нём встречаются все числа от $\min\{p_l, \dots, p_r\}$ до $\max\{p_l, \dots, p_r\}$. Посчитайте количество хороших подотрезков.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leq n \leq 300\,000$) — длина перестановки.

Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — элементы перестановки. Гарантируется, что если $i \neq j$, то $p_i \neq p_j$.

Формат выходных данных

Выведите единственное число — количество хороших подотрезков.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 4 5	15
6 1 2 5 3 6 4	10
1 1	1

Задача В. Хорошие подотрезки hard

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дана перестановка p чисел $1, \dots, n$. Назовём подотрезок $[l, r]$ хорошим, если на нём встречаются все числа от $\min\{p_l, \dots, p_r\}$ до $\max\{p_l, \dots, p_r\}$.

Даны q запросов. В каждом запросе даны два числа l и r . Требуется посчитать количество хороших подотрезков на отрезке $[l, r]$, то есть количество таких $l \leq a \leq b \leq r$, что подотрезок $[a, b]$ хороший.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leq n \leq 300\,000$) — длина перестановки.

Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — элементы перестановки. Гарантируется, что если $i \neq j$, то $p_i \neq p_j$.

Третья строка содержит единственное целое число q ($1 \leq q \leq 300\,000$) — количество запросов.

Следующие q строк содержат по два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$) — отрезок i -го запроса.

Формат выходных данных

Для каждого запроса выведите количество хороших подотрезков на отрезке запроса.

Примеры

стандартный ввод	стандартный вывод
1	1
1	
1	
1 1	
5	1
1 3 2 5 4	2
15	5
1 1	6
1 2	10
1 3	1
1 4	3
1 5	4
2 2	7
2 3	1
2 4	2
2 5	4
3 3	1
3 4	3
3 5	1
4 4	
4 5	
5 5	

Задача С. Джокер

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Джокер вернулся в Готэм-Сити для осуществления очередного злодейского плана. В Готэм-Сити есть N перекрёстков (пронумерованных от 1 до N) и M дорог (пронумерованных от 1 до M). Каждая улица соединяет два различных перекрёстка, и любые два перекрёстка соединены не более одной улицей.

Для своего злодейского плана, Джокеру нужно использовать нечётное количество улиц, которые образуют цикл. Формально, для перекрёстка S и чётного натурального k , должна существовать такая последовательность перекрёстков S, s_1, \dots, s_k, S , что есть улицы, соединяющие перекрёстки (a) S и s_1 , (b) s_k и S , и (c) s_{i-1} и s_i для каждого $i = 2, \dots, k$.

Однако, полиция патрулирует улицы Готэм-Сити. Каждый день i , она наблюдает за конкретным подмножеством улиц с последовательными номерами j : $l_i \leq j \leq r_i$. Эти улицы под наблюдением не могут быть использованы Джокером для своего плана в этот день. К несчастью для полиции, у Джокера есть шпионы среди Отделения Полиции Готэм-Сити; они доносят ему, в какие дни за какими улицами ведётся наблюдение. Теперь Джокер хочет узнать для некоторого количества дней, может ли он повернуть свой план в каждый из этих дней или нет. План может быть осуществлён, если есть цикл с нечётным количеством улиц, которые не находятся под наблюдением в данный день.

Формат входных данных

Первая строка ввода содержит три целых числа N, M и Q ($1 \leq N, M, Q \leq 200\,000$): количество перекрёстков, улиц и интересующих дней, соответственно. Следующие M строк содержат описание улиц. j -тая из этих строк ($1 \leq j \leq M$) содержит номера двух улиц u и v ($u \neq v$), что означает, что улица j соединяет эти два перекрёстка. Гарантируется, что любые два перекрёстка соединены не более одной улицей. Каждая из следующих Q строк содержит по двум целым числам l_i и r_i , что означает, что в день i ($1 \leq i \leq Q$) под наблюдением полиции находятся все улицы j с номерами $l_i \leq j \leq r_i$.

Формат выходных данных

Выведите Q строк. i -тая строка ($1 \leq i \leq Q$) должна содержать «YES», если Джокер может осуществить план в день i , и «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
6 8 2 1 3 1 5 1 6 2 5 2 6 3 4 3 5 5 6 4 8 4 7	NO YES
2 1 1 1 2 1 1	NO

Задача D. Минимизируй количество инверсий

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана перестановка p длины n .

Можно выбрать любую её подпоследовательность, удалить её из перестановки и приписать в начало перестановки в том же порядке.

Для каждого k от 0 до n найдите минимальное количество инверсий в перестановке, которое можно получить, выбрав подпоследовательность длины ровно k .

Формат входных данных

В первой строке задано одно целое число t ($1 \leq t \leq 50\,000$) — количество наборов входных данных.

В первой строке каждого набора вводится число n ($1 \leq n \leq 5 \cdot 10^5$) — длина перестановки.

Во второй строке каждого набора вводится перестановка p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$).

Гарантируется, что сумма n не превосходит $5 \cdot 10^5$.

Формат выходных данных

Для каждого набора выведите $n + 1$ число, где i -е число — ответ на задачу для длины подпоследовательности, равной $i - 1$.

Пример

стандартный ввод	стандартный вывод
3	0 0
1	4 2 2 1 4
1	5 4 2 2 1 5
4	
4 2 1 3	
5	
5 1 3 2 4	

Замечание

Во втором наборе:

- Для длины 0: $[4, 2, 1, 3] \rightarrow [4, 2, 1, 3]$: 4 инверсии.
- Для длины 1: $[4, 2, 1, 3] \rightarrow [1, 4, 2, 3]$: 2 инверсии.
- Для длины 2: $[4, 2, 1, 3] \rightarrow [2, 1, 4, 3]$ или $[4, 2, 1, 3] \rightarrow [1, 3, 4, 2]$: 2 инверсии.
- Для длины 3: $[4, 2, 1, 3] \rightarrow [2, 1, 3, 4]$: 1 инверсия.
- Для длины 4: $[4, 2, 1, 3] \rightarrow [4, 2, 1, 3]$: 4 инверсии.

Задача E. Ним?

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Есть n кучек с камнями. i -я кучка содержит a_i камней. Два игрока играют в игру:

- Первый игрок может выбрать любое ненулевое количество кучек с камнями и из каждой из них забрать по x камней. Очевидно, что игрок не может забрать x камней из кучки, в которой нет такого количества камней.
- Второй игрок аналогично может выбрать любое ненулевое количество кучек и из каждой из них забрать по y камней.

Игрок, который не может сделать ход проигрывает. Определите победителя игры.

Формат входных данных

Первая строка входных данных содержит три целых числа n, x, y ($1 \leq n \leq 2 \cdot 10^5, 1 \leq x, y \leq 10^9$) — количество кучек с камнями, а так же сколько камней должны забирать игроки на своём ходу.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — количество камней в кучках.

Формат выходных данных

Выведите «First», если побеждает первый игрок, и «Second» иначе.

Примеры

стандартный ввод	стандартный вывод
2 1 1 3 3	First
2 1 2 3 3	Second

Задача F. Игра на дереве

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано дерево на n вершинах. Изначально в i -й вершине есть a_i камней.

Два игрока играют в следующую игру. Первый игрок выбирает вершину и ставит на неё фишку. Далее два игрока по очереди (начиная с первого игрока) делают следующие действия:

1. Убрать один камень из вершины, в которой сейчас находится фишка.
2. Передвинуть фишку в соседнюю вершину.

Игрок, на ходу которого фишка оказалась в вершине, в которой не осталось камней — проигрывает.

Определите все вершины, в которые первый игрок может поставить фишку в самом начале игры, чтобы гарантировать себе победу.

Формат входных данных

Первая строка содержит единственное целое число n ($2 \leq n \leq 3000$) — количество вершин в дереве.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — количество камней в каждой из вершин.

Следующие $n - 1$ строк содержат описание рёбер дерева. i -я строка содержит два целых числа v и u ($1 \leq v, u \leq n$) — вершины, которые соединяет i -е ребро.

Формат выходных данных

Выведите все вершины, начиная с которых, побеждает первый игрок, в отсортированном порядке.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 1 2 2 3	2
5 5 4 1 2 3 1 2 1 3 2 4 2 5	1 2
3 1 1 1 1 2 2 3	