# Задача A. Хоровод

Для начала разберём случай $x = y$. Понятно, что максимальное количество пар соседей разного пола достигается при чередовании мальчиков и девочек: `BGBGBG...BG`. В данном случае количество пар равно $x + y$. Минимальное количество пар достигается, если все мальчики идут подряд, а потом все девочки идут подряд: `BB...BBGG...GG`, тогда количество пар равно 2.

Заметим, что количество пар соседей разного пола всегда чётно. Чтобы доказать это, можно рассмотреть блок подряд идущих мальчиков, с ним образуется ровно 2 пары с девочками (слева и справа). Если просуммировать это по всем блокам мальчиков, то мы получим ровно количество пар соседей разного пола, и оно будет чётным.

Теперь, чтобы получить меньше, чем $x + y$ пар соседей разного пола, можно начать объединять детей одного пола в блоки. Например:

- `GBGBGBGB` — 8 пар;
- `GGBBGBGB` — 6 пар;
- `GGGBBBGB` — 4 пары;
- `GGGGBBBB` — 2 пары.

То есть, мы каждый раз берём следующего мальчика и девочку и добавляем в блоки подряд идущих мальчиков и девочек, уменьшая количество пар на 2. Так можно получить любое чётное количество пар от 2 до $x + y$.

Теперь перейдём к случаю $x \neq y$. Понятно, что у нас не могут просто чередоваться мальчики и девочки. Однако максимальное количество пар достигается, когда они почти чередуются: `BGBGBG...BGBGGGGG` (в случае, если девочек больше). Количество пар соседей разного пола в этом случае равно $2 \cdot \min(x, y)$. Чтобы получить какое-то промежуточное количество пар (между 2 и $2 \cdot \min(x, y)$), можно так же, как описано выше, объединять мальчиков и девочек в блоки.

# Задача B. Суммы из отрезка

Сначала разберём отдельный случай: с помощью суммы чисел на отрезке можно представить любое число, если $l = 1$. В противном случае в виде суммы нельзя представить число 1. То есть ответ — $-1$ тогда и только тогда, когда $l = 1$.

Далее, число $k$ представимо в виде суммы $x$ чисел из отрезка, если выполняется $l \cdot x \leqslant k \leqslant r \cdot x$. А значит число не представимо, если для какого-то $x$ имеем $r \cdot x < k < l \cdot (x+1)$, то есть $x$ слагаемых — мало, а $x + 1$ — много. Значит, нам нужно найти максимальный «пропуск» такого вида.

Имеем неравенство, где $x$ — количество слагаемых: $r \cdot x + 1 < l \cdot (x + 1)$ (так как между ними должно быть хотя бы одно число). Решая его, получаем $x < \frac{a-1}{b-a}$. А так как нам нужен максимальный целый $x$, то он будет равен $\lfloor \frac{a-2}{b-a} \rfloor$. Мы нашли максимальный «пропуск», а максимальное число из него равняется $l \cdot (\lfloor \frac{a-2}{b-a} \rfloor + 1) - 1$.

# Задача C. Крайний Банк

Пусть мы знаем все пары покупка-продажа, как трейдер дошел до состояния, описанного во входных данных. Заметим, что если есть пара вида $(i, j)$ и в момент $j$ у нас была какая-то незакрытая сделка, в которой стоимость покупки больше, чем $a_i$, но при этом меньше $a_j$, то мы можем поменять моменты покупки в этих парах и все останется корректным.

Из этого замечания сразу же следует жадный алгоритм для решения задачи. Давайте идти по моментам времени и если мы находимся в моменте, где трейдер продавал, то будем ставить ему в пару максимальные незанятые моменты покупки, стоимость товара в которых не превышает текущей стоимости продажи. Таким образом мы либо получим корректное разбиение на пары, либо в какой-то момент столкнемся с противоречием (не будет подходящих моментов покупки), что значит, что трейдер никак не мог исполнить условие, для того чтобы быть потенциальным мошенником.

Для эффективной реализации данного решения предлагается хранить multiset с парами вида $(a_i, c_i)$, означающих, что у нас есть «незакрытая» покупка по цене $a_i$, в которую мы купили $c_i$

бипок. При продаже нужно делать lower bound и брать предыдущий индекс, после чего корректно пересчитать multiset, и так пока мы не закроем всю продажу. Такое решение работает за $O(n \log n)$.

# Задача D. Максимальная попарная разница

Если раскрыть все модули в $|a_1 - b_{p_1}| + |a_2 - b_{p_2}| + \ldots + |a_n - b_{p_n}|$, то $n$ чисел из $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ войдут в сумму со знаком плюс, а оставшиеся $n$ чисел со знаком минус.

Поэтому, максимальный счёт точно не может быть больше чем «сумма $n$ максимальных чисел среди $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ минус сумма $n$ минимальных чисел среди $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$». И оказывается, что такого счёта всегда можно достичь.

Давайте отсортируем все числа $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ и посмотрим: сколько чисел из массива $a$ окажется в первой половине отсортированного массива, пусть $x$. Тогда чисел из массива $b$ в первой половине всего $n - x$ штук. А во второй половине: $n - x$ чисел из массива $a$, и $x$ чисел из массива $b$. Поэтому мы можем образовать $n$ пар: «число из массива $a$, число из массива $b$» так, что в каждой паре по одному числу из каждой половины отсортированного массива. И счёт в таком случае, будет в точности равен «сумма $n$ максимальных чисел среди $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ минус сумма $n$ минимальных чисел среди $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$».

А число способов это сделать будет $x! \cdot (n - x)!$, где $k! = 1 \cdot 2 \cdot \ldots \cdot k$. Так как нужно независимо сопоставить $x$ наименьшим числам из $a$ — $x$ наибольших чисел из $b$, и независимо $n - x$ наибольшим числам из $a$ — $n - x$ наименьших чисел из $b$.

При этом при любом другом способе, хотя бы одна из пар $(a_i, b_{p_i})$ будет содержать пару чисел из одной и той же половины отсортированного массива, и, так как все числа различны, в таком случае счёт будет не максимален.

Асимптотика решения: $O(n \log n)$, из-за сортировки массива.

# Задача E. Рекордное удаление

**1 группа**

Для каждого запроса промоделируем процесс удаления. На каждом этапе будем искать последовательность рекордов и запоминать элементы, которые мы уже удалили. Получаем асимптотику $O(nk)$ на один запрос и $O(qnk)$ итоговую.

**2 группа**

В данной группе перестановка отсортирована по убыванию. Рассмотрим произвольный запрос. Заметим, что на каждом этапе удаления длина последовательности рекордов не превосходит 1. Если мы совершаем $k$ этапов удаления, то ответ это минимум из $k$ и количества чисел на суффиксе.

**3 группа**

Сохраним для каждой позиции перестановки все запросы, суффиксы которых начинаются в этой позиции. Будем перебирать индекс перестановки от конца к началу и отвечать сразу на все запросы для одной позиции. Для решения этой группы заведем стек рекордов. При переходе от позиции $i$ к позиции $i - 1$ удалим с верхушки стека все элементы, меньшие $a_{i-1}$ и положим $a_{i-1}$ в стек (т.к. $a_{i-1}$ точно окажется в последовательности рекордов на первом этапе удаления). Чтобы узнать длину последовательности рекордов, посмотрим на размер стека. Каждый элемент попадает в стек ровно один раз и будет удален из стека не более одного раза, поэтому итоговая асимптотика $O(n + q)$.

**4 группа**

Улучшим решение для предыдущей группы. Вместо одного стека будем поддерживать два — в первом будет храниться последовательность рекордов для текущего суффикса, а во втором последовательность рекордов, если совершить 1 этап удаления. Посмотрим на те элементы, которые были удалены из первого стека при переходе от $i$ к $i - 1$. Заметим, что эти элементы неизбежно окажутся в последовательности рекордов после первого этапа удаления. Мы добавим их во второй стек, но перед этим нужно понять, какие элементы могут перестать входить в последовательность рекордов после первого этапа удаления. На самом деле, это все те элементы, что не превосходят максимум среди добавляемых элементов. Удалим все эти значения из второго стека и положим новые. Это легко сделать, так как значения в стеках идут в возрастающем порядке. Можно заметить, что после такой процедуры значения в стеках по-прежнему будут идти в возрастающем порядке.

**5 группа**

Воспользуемся тем, что $k$ в любом запросе не превосходит 20. Обобщим идеи для прошлых групп и заведем сразу 20 стеков. В $i$-м стеке будет лежать последовательность рекордов после $i-1$ этапа удалений. При добавлении нового элемента обновим первый стек, а дальше будем идти стекам, начиная со второго, и обновлять их через те элементы, которые были выброшены из предыдущего стека. Если на каком-то шаге мы не выбросили ни одного элемента, прекратим этот процесс. Для ответа на запрос посчитаем сумму размеров первых $k$ стеков. Каждый элемент добавится в каждый стек и удалится из каждого стека не более одного раза, поэтому итоговая асимптотика $O(nk + qk)$.

# Задача F. Дорожный патруль

Заметим, что оптимальными значениями для ограничения являются числа $a_i$, поэтому таких значений не более $n$. Давайте отсортируем машины по убыванию скорости и будем при добавлении очередной машины заново считать ответ.

При $t \geqslant 3000$ заметим, что мы остановим не более $O(\frac{n}{t})$ машин, поэтому мы можем сложить позиции всех машин со скоростью больше данной в set, а следующую, которую нужно остановить, искать с помощью lower bound. Такое решение работает за $O(n\frac{n}{t}\log n)$.

В общем случае, давайте разделим весь массив на блоки по $\sqrt{n}$, для каждого элемента будем пересчитывать первую машину **вне его блока**, которую мы остановим после него. Аналогично будем пересчитывать сумму $a_i$ всех машин между данной и следующей из другого блока, а также число таких машин. Заметим, что при добавлении новой машины мы можем пересчитать весь блок за его размер (т.е. за $O(\sqrt{n})$) проходом справа налево.

Осталось уметь пересчитывать ответ, используя такую структуру. Достаточно поддерживать первую машину, которую мы остановим, после чего, начиная с неё, переходить к машинам в следующих блоках, с помощью того, что мы насчитали. При таком процессе нужно поддерживать число остановленных машин и сумму их скоростей, после чего для данного ограничения скорости можно восстановить ответ. Такое решение работает за $O(n\sqrt{n})$.