Задача A. Случайный порядок

Для решения выпишем индексы каждого числа в массив $i_1, \cdots, i_n$. Тогда требуется найти такое максимально возможное k, что $max(i_1, \cdots, i_k) < min(i_{n-k+1}, \cdots, i_n)$.

Это можно сделать, постепенно увеличивая k и поддерживая максимум на префиксе и минимум на суффиксе до того момента, пока условие выше выполняется. Данное решение работает за $\mathcal{O}(n)$.

Задача B. Точка покоя

Для решения рассмотрим следующую задачу: даны длины векторов $a_1, \cdots, a_n$, необходимо определить, можно ли их направить так, чтобы в сумме получился нулевой вектор.

Тогда, если максимальный элемент a больше суммы всех остальных элементов, то этот самый большой вектор невозможно привести в ноль.

Иначе в сумме может получиться нулевой вектор – доказать это можно, например, следующим образом. Пусть изначально все остальные векторы направлены вдоль вектора с максимальной длиной и упорядочены. Тогда возьмём первый из них и непрерывно повернём на 180, при этом так направляя все следующие векторы во время движения, что они по-прежнему сонаправлены и их конец лежит на прямой, проходящей через продолжение вектора с максимальной длиной.

В итоге мы получим все векторы, направленные в сторону вектора с максимальной длиной (изначально все векторы, кроме вектора с максимальной длиной, были направлены в противоположную сторону). Но все движения были непрерывны, а изначально сумма длин остальных векторов была не меньше длины максимального вектора. Это означает, что в некоторый момент времени сумма всех векторов в точности равнялась нулевому вектору.

Вернёмся к исходной задаче. Рассмотрим отрезок $[l_i; r_i]$ с максимальным l_i. Затем рассмотрим сумму всех остальных r_j $restsum$. Тогда, если $l_i > restsum$, то сумма длин любого набора векторов не даст нулевой вектор, поскольку любое другое значение отрезка i только увеличивает максимум, а любые другие значения остальных отрезков только уменьшают оставшуюся сумму.

Иначе можно сделать длину отрезка i равной l_i, а у всех остальных отрезков $len_j = min(r_j, l_i)$ – это число всегда находится в отрезке $[l_j; r_j]$, поскольку $l_j \leqslant l_i$. Также сумма этих чисел не меньше l_i, поскольку, если хоть раз $len_j = l_i$, то условие суммы сразу выполнено, иначе оно выполнено, поскольку сумма оставшихся r_j не меньше l_i.

Решение работает за $\mathcal{O}(n)$.

Задача C. Альтернативные миры I

Для решения задачи заметим несколько особенностей оптимального разбиения.

1) Если медиана множества S_1 равна x_1 и медиана множества S_2 равна $x_2 \geqslant x_1$, то $x = med(S_1 \cup S_2) \in [x_1; x_2]$.

x не может быть больше x_2, поскольку добавляя элементы S_1 к S_2, элементов, которые больше x_2, меньше, чем тех, которые не больше x_2.

Аналогичными рассуждениями получаем, что x не может быть меньше x_1.

2) Существует оптимальное разбиение, в котором в каждом множестве содержится не более одного неотрицательного числа.

Действительно, если есть множество длины 2 из двух неотрицательных чисел, то его можно разбить на 2 множества, не уменьшая сумму медиан.

Иначе (длина не менее 3) можно убрать из данного множества первый и последний элемент (который всегда будет положительным) в отдельное множество. Медиана исходного множества не изменится, но в то же время медиана нового множества будет не меньше 0. Значит, сумма медиан не уменьшилась.

Многократно повторяя рассуждения выше (например, по индукции), получим, что 2 верно.

Теперь опишем одно из оптимальных решений.

Если неотрицательных чисел не меньше, чем количество отрицательных чисел, то сделаем для каждого неотрицательного числа отдельное множество, и затем оставшиеся отрицательные числа по одному добавим в некоторые множества.

Тогда отрицательные числа не дадут никакого вклада, а неотрицательные все просуммируются – это является максимально возможной суммой медиан.

Если отрицательных чисел больше, то из 2 следует, что если неотрицательных чисел cnt штук, то только cnt отрицательных чисел можно добавить к ним, чтобы они не вносили никакого вклада. Очевидно, что оптимально добавлять самые малые отрицательные числа – так создадим cnt множеств из двух чисел, одно из которых неотрицательно, а второе отрицательное.

При этом из 1 следует, что оставшиеся числа выгодно оставить в одном множестве.

Таким образом, мы получили оптимальное разбиение для каждого случая, из которого несложно получить ответ.

Задача D. Альтернативные миры II

Для решения задачи заметим несколько особенностей оптимального разбиения.

1) Если медиана множества S_1 равна x_1, и медиана множества S_2 равна $x_2 \geqslant x_1$, то $x = med(S_1 \cup S_2) \in [x_1; x_2]$.

Данное утверждение доказывается схожим образом с соответствующим утверждением предыдущей задачи.

Также из него аналогичным образом следует, что только 1 множество может иметь отрицательную медиану (иначе эти 2 множества можно объединить в одно).

2) Покажем, что из множеств с неотрицательной медианой только одно содержит более чем 1 элемент (то есть что оптимальное разбиение с таким свойством существует).

Для начала, если в множестве с положительной медианой больше 1 элемента, то это множество нечётного размера, иначе можно отделить один из элементов a, b, вносящих вклад, в отдельное множество. Тогда сумма медиан данных элементов поменяется с $(a + b)/2$ до $a + b$.

Также в множестве с положительной медианой из более чем 1 числа все числа, меньшие медианы, отрицательные, – иначе данные положительные числа можно отправить в отдельные множества, что не уменьшит сумму медиан.

Но тогда, если есть 2 множества с положительной медианой и более чем из 1 числа каждое, с медианами $a, b(a \leqslant b)$, то можно выделить b в отдельное множество, а все оставшиеся числа слить в 1 множество, медиана которого будет равна a — то есть сумма медиан при таком преобразовании не изменится.

Что и требовалось доказать для 2.

3) Пусть множество с отрицательной медианой называется A и имеет длину m, а множество с количеством положительных чисел, большим 1, называется B и имеет размер $k = 2x+1(x \geqslant 0)$. Докажем, что существует оптимальное разбиение, в котором существует лишь одно из этих множеств.

Докажем от противного. Пусть такие множества есть.

Тогда в множестве B ровно $x + 1$ неотрицательных чисел (что ранее было доказано) и x отрицательных. Кроме того, поскольку все эти числа не вносят вклад в медиану, выгодно брать x самых малых отрицательных. Только меньшее неотрицательное число вносит вклад, поэтому выгодно взять только первые $x + 1$ положительных.

Но тогда в множестве A все отрицательные числа не меньше отрицательных чисел в множестве B, и аналогично для неотрицательных чисел.

Это означает, что можно, например, взять самое малое и самое большое число из B и добавить их в A, и сумма медиан не изменится. Тогда, повторяя этот процесс многократно, множество A будет состоять только из одного элемента.

4) Теперь покажем, что если есть только множество A или B, то они состоят из некоторого префикса чисел в отсортированном массиве.

Действительно, если B существует, то оно определяется однозначно и состоит из префикса (доказано ранее).

Иначе существует A, состоящее из всех отрицательных чисел и каких-то неотрицательных. Но тогда в A входят минимально возможные неотрицательные числа (иначе можно заменить числа A меньшими положительными в отдельных множествах и не уменьшить сумму медиан).

Таким образом, достаточно проверить лишь ситуации, в которых некоторый префикс отсортированного изначального массива это отдельное множество, а остальные элементы – являются множествами из 1 элемента. Максимальная сумма таких медиан и будет ответом.

Задача E. Тёмный лабиринт

Будем рассматривать все вершины данного массива c как компоненту двусвязности, которую будем считать вершиной. Изначально $c = [1]$, и это верно.

1) Пусть в графе существует цикл длины не более $k + 1$, который проходит через данную компоненту двусвязности, рассматриваемую как вершину. Тогда утверждается, что все вершины этого цикла можно добавить в c.

Действительно, можно поставить лампу в компоненту c, а далее, если длина цикла не более k, поставить лампы в каждую его вершину. Иначе, если длина цикла равна $k + 1$, то достаточно поставить лампы в его следующие после c $k-1$ вершин, поскольку тогда только одна вершина будет не освещённой, что означает, что по туннелям, связанным с ней, можно проходить.

Тогда, многократно проходя данный цикл, можно добавить все его вершины в c, а из того, что это был цикл, следует, что c останется двусвязной компонентой.

После этого можно забрать все лампы.

2) Вершина n до сих пор не в c, но любой цикл, проходящий через c (рассматриваемую как одну вершину), содержит минимум $k + 2$ вершины.

Тогда ни в какие вершины, кроме c, попасть невозможно (с возможностью сохранить лампы). Это верно, поскольку для того, чтобы начать путь по вершинам, не входящим в c, нужно поставить лампу в c и затем сделать шаг из компоненты c. Но после этого Оливер не сможет вернуться в компоненту c.

Докажем от противного. Пусть возможно попасть в вершину u, из которой исходит ребро в c. Рассмотрим первый раз, когда мы попадаем в такую вершину. Тогда до этого Оливер не заходил в вершины, ведущие в c, а значит, не мог увеличивать c или собирать лампы. То есть в вершину u можно попасть, используя k ламп и не собирая их.

Пусть $d(v)$ это минимальное число рёбер, которое нужно пройти из c, чтобы попасть в вершину v. Поскольку любой цикл через c содержит минимум $k+2$ рёбер, а u имеет ребро в c, то $d(u) \geqslant k+1$.

По индукции докажем, что, используя k ламп, можно попасть в вершины v, для которых $d(v) \leqslant k$.

База индукции: $k = 1$ - тогда, чтобы начать путь, есть только 1 вариант – поставить лампу в вершину c – тогда можно будет попасть только в вершины, в которые есть ребро из c напрямую.

Переход индукции: пусть утверждение верно для k – докажем его для $k + 1$.

Докажем от противного – пусть можно попасть в вершину v, где $d(v) = k + 2$. Пусть это первая вершина с $d(v) = k+2$, в которую можно попасть. Но тогда предыдущая вершина u имеет $d(u) = k+1$, а также для того, чтобы попасть в v, необходимо было поставить лампу в u. Рассмотрим первый момент, когда Оливер мог зайти в вершину u – до этого в u не было лампы, которую он далее туда поставит, но он смог попасть в вершину u, для которой $d(u) = k + 1$, используя всего k ламп, – противоречие.

Это значит, что в вершины, где $d(v) \geqslant k + 2$, попасть невозможно, – что и требовалось доказать.

Таким образом, в рассматриваемую вершину u, для которой $d(u) \geqslant k + 1$, невозможно попасть, используя k ламп.

Таким образом, мы получаем алгоритм решения – необходимо находить кратчайшие циклы из текущей компоненты c и, пока их длина не более $k + 1$, сжимать их, а в конце проверить, что $n \in c$.

Для поиска кратчайшего цикла можно использовать bfs из вершины c. Поскольку сжатие может происходить не более n раз и bfs занимает $\mathcal{O}(m)$ времени, то решение работает за $\mathcal{O}(mn)$.

Задача F. Сортировка Таноса

Рассмотрим, какие подотрезки исходного массива можно получить, выполняя сортировку Таноса. Можно заметить, что для исходного массива мы либо завершаем сортировку сразу (используем весь массив), либо переходим в одну из частей деления массива пополам (насколько это возможно). То есть все возможные итоговые подотрезки исходного массива сами образуют древовидную структуру, на подобие дерева отрезков (глубины $\mathcal{O}(\log n)$).

Пусть некоторый отрезок на глубине h (для корня $h = 0$) отсортирован, и все его предки не отсортированы. Тогда в него можно перейти с вероятностью 2^{-h}, и если он имеет длину len, то даёт вклад в матожидание, равный $len \cdot 2^{-h}$. Тогда для поддержания ответа будем при каждом изменении пересчитывать вклад всех вершин.

Тогда при изменении значения в некотором индексе пройдём по всем $\mathcal{O}(\log n)$ вершинам дерева отрезков и для каждой из них проверим, является ли теперь соответствующий ей отрезок отсортированным.

Далее необходимо аккуратно пересчитать ответ — если новый самый глубокий отсортированный отрезок выше предыдущего на этом пути, то нужно убрать вклад со всех отрезков ниже данного (в каждую соответствующую вершину можно добраться из рассматриваемых $\mathcal{O}(\log n)$ вершин за 1 ребро, поскольку изменилось значение лишь одного элемента). Если новый самый глубокий отсортированный отрезок ниже предыдущего, то нужно учесть отрезки ниже по пути, аналогично случаю выше.

Таким образом, при каждом изменении рассматривается лишь $\mathcal{O}(\log n)$ вершин.

Для того чтобы для каждой из $\mathcal{O}(\log n)$ вершин на пути определить, является ли соответствующий ей отрезок отсортированным, можно построить массив разностей соседних элементов и дерево отрезков на нём, а далее бинарным подъёмом по этому дереву найти наибольший отсортированный отрезок, используя который можно рассчитать наиболее глубокую вершину, соответствующую отсортированному отрезку. Данный массив разностей можно также поддерживать в самом дереве отрезков.

Описанное выше решение будет работать за $\mathcal{O}(n + q \log n)$.

Задача G. Реконструкция

Нам задан граф, показывающий ближайшие к каждому числу число (его индекс). Можно заметить, что этот граф всегда будет состоять из компонент связности, каждая из которых это цикл длины 2, на каждую из вершин которого может указывать 1 ребро из некоторой вершины, на которую также может указывать максимум 1 ребро и так далее.

Тогда для проверки существования достаточно проверить, что заданный граф имеет соответствующую структуру.

Также очевидно, что для минимизации максимума a нужно, чтобы расстояние между элементами было минимально возможным. Поэтому между элементами каждого цикла должно быть расстояние 1, далее до следующей вершины в каждом направлении 2 и так далее, увеличивая расстояние на 1.

Так мы получим набор компонент, к которым с каждого конца есть 2 расстояния a_i, b_i. Тогда между компонентами, которые находятся на числовой прямой рядом друг с другом, должно быть расстояние $max(x, y) + 1$, если 2 компоненты имеют расстояния x, y между граничащими вершинами и предыдущими точками в своих компонентах.

То есть для восстановления ответа необходимо восстановить оптимальный порядок компонент и их направление (с какой из сторон на конце будет a, а с какой b).

При этом нижняя часть первой компоненты x и последняя верхняя y не вносят никакого вклада.

Однако допустим, что задача циклическая и они вносят дополнительный вклад $max(x, y) + 1$. Тогда её можно решить жадным алгоритмом: возьмём компоненту с максимальным числом среди всех a_i, b_i. Поскольку оно точно идёт во вклад, то выгодно использовать с этим максимальным числом следующее максимальное число из всех оставшихся. Таким образом можно объединить 2 найденные блока в один и добавить их вклад и итоговую сумму.

Повторив этот процесс многократно, получим оптимальное решение для циклической задачи. Чтобы вернуться к текущей задаче, достаточно убрать разделение по максимальному числу и восстановить ответ.

Асимптотика решения: $\mathcal{O}(n \log n)$.