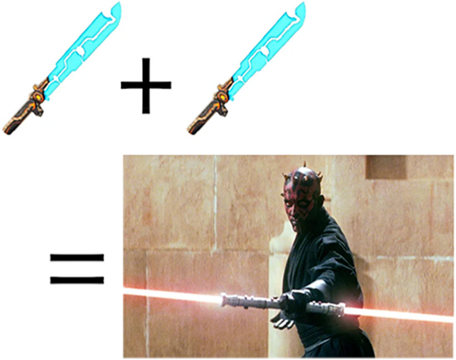
Олимприада имени Келдыша 2023

Разбор задач

18 июня 2023 г.

«A. Максимальная прочность»



- Идея задачи — Жюри
- Разработка задачи — Александр Понкратов, СУНЦ МГУ

Постановка задачи

- Дан отрезок $[L, R]$, $1 \leqslant L \leqslant R < 10^{10^5}$
- Среди всех пар чисел на отрезке найти максимальное значение суммы модулей разностей цифр в их десятичной записи на каждой позиции

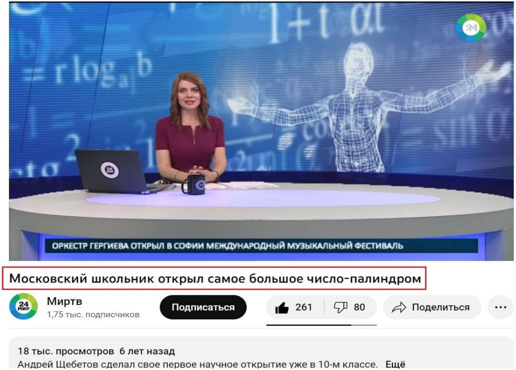
Ключевая идея

- Максимальную разность в каждом разряде можно получить, если в одном числе стоит 0, а в другом 9
- Числа L и R можно представить как их наибольший общий префикс, разряд k, в котором значения различаются, и остальные цифры
- После разряда k можно ставить любые цифры

Полное решение

- Дополним L ведущими нулями при необходимости
- Оба числа представимы как $L = \overline{l_1 l_2 \ldots l_n}$ и $R = \overline{r_1 r_2 \ldots r_n}$
- Найдем первый индекс k такой, что $l_k \neq r_k$
- Ответ равен $(r_k - l_k) + 9 \cdot (n - k)$
- Ответ достигается при $A = \overline{l_1 l_2 \ldots l_k \underbrace{99 \ldots 9}_{n-k}}$ и $B = \overline{r_1 r_2 \ldots r_k \underbrace{00 \ldots 0}_{n-k}}$

«B. Игра с переворотом»



- Идея задачи — Лепешов Всеволод, ВШЭ
- Разработка задачи — Лепешов Всеволод, ВШЭ

Постановка задачи

- Даны две строки одинаковой длины.
- Двое играют в игру: первый в свой ход заменяет любой символ любой из строк, второй в свой ход разворачивает любую из строк, ходят по очереди.
- Игра идёт пока строки не равны. Цель первого закончить игру как можно раньше, цель второго: как можно позже. Нужно узнать сколько ходов продлится игра.

Боб переоценён?

- Анализируя семплы может сложиться ощущение, что выбор хода со стороны Боба ничего не решает
- И... это действительно так! Покажем это:
- Разворот строки **два** раза ничего не меняет $\rightarrow$ нас интересует только чётность
- Если суммарно Боб сделал чётное число ходов, значит чётность количества ходов сделанных Бобом с строкой S совпадает с чётностью количества ходов сделанных Бобом с строкой T $\rightarrow$

Стратегия Алисы

- $\to$ Пары символов на одинаковых позициях будут $(s_1, t_1), (s_2, t_2), \ldots, (s_n, t_n)$, возможно в перевёрнутом порядке, но нам это не важно.
- Аналогично с нечётным числом ходов, но ровно одна из строк по итогу будет развёрнута, и пары будут: $(s_1, t_n), (s_2, t_{n-1}), \ldots, (s_n, t_1)$.
- Алиса может выбрать одну из двух стратегий: сделать $s_1 = t_1, s_2 = t_2, \ldots, s_n = t_n$, когда количество ходов Боба будет чётным,
- Или же сделать $s_1 = t_n, s_2 = t_{n-1}, \ldots, s_n = t_1$, когда количество ходов Боба будет нечётным.

Финальное решение

- Посчитаем cnt — количество индексов, в которых отличаются S и T, и cnt_{rev} — количество индексов, в которых отличаются S и $\text{rev}(T)$.
- Для первой стратегии игра продлится $2 \cdot cnt - cnt\%2$ ходов.
- Для второй стратегии игра продлится $2 \cdot cnt_{rev} - (1 - cnt_{rev}\%2)$ ходов.
- Ответом будет минимум из этих двух значений. Также нужно быть аккуратным в случае $cnt_{rev} = 0$, в таком случае для второй стратегии игра длится 2 хода.

«C. Опрос на уроке»

- Идея задачи — Лазарев Никита, ВШЭ
- Разработка задачи — Дацковский Алексей, ВШЭ

Постановка задачи

- Дан набор отрезков
- Пусть дан набор чисел A. Тогда счётом отрезка $[l, r]$ будет количество x из A таких, что $x \in [l, r]$, минус количество x из A таких, что $x \notin [l, r]$.
- Нужно найти набор чисел A такой, что разница между максимальным и минимальным счётом среди данных отрезков будет как можно больше, и вывести эту разницу.

Сведение задачи к отрезкам

- Зафиксируем учеников, у которых будет в итоге самая высокая и самая низкая рука.
- Чтобы достичь максимальной разности, можно спросить все темы, которые знает ученик с высокой рукой.
- Тогда второй ученик поднимет руку для каждой темы на пересечении их отрезов, и опустит на каждую тему, которую знает только первый.

Сведение задачи к отрезкам

- Итого нужно решить такую задачу: Дан набор отрезков. Нужно найти 2 отрезка a и b таких, что значение $|a| - |a \cap b|$ максимально, а ответом будет $2 \cdot (|a| - |a \cap b|)$.

Перебор за $\mathcal{O}(n^2)$

- Зафиксируем 2 отрезка — ученика с самой высокой и самой низкой рукой.
- Посчитаем, какая разница будет между их руками, если назвать темы, которые знает только первый.
- Среди всех пар найдём самую большую разность и выведем её.

Пересечение отрезков

- Отрезок b может пересекать отрезок a четырьмя способами:
- пересекать начало a
- пересекать конец a
- полностью входить в a
- не пересекать вообще.

Пересечение отрезков

- При этом, если в ответе отрезок b пересекает начало a, то в качестве отрезка b можно выбрать отрезок с минимальным правым концом — пересечение такого отрезка с a будет не больше пересечения b с a.
- Аналогично, для правого края можно рассмотреть отрезок с максимальным левым концом.

Пересечение отрезков

- Если же в ответе один отрезок вложен в другой, то в качестве внутреннего отрезка можно рассмотреть самый короткий отрезок в наборе.
- Если отрезки в ответе не пересекаются, то отрезок не пересекается с одним из «краевых» отрезков.

Полное решение

- Таким образом, чтобы для данного отрезка найти отрезок, с которым у него минимальное пересечение, нужно проверить 3 кандидата:
- самый короткий отрезок в наборе
- отрезок с минимальным правым концом
- отрезок с максимальным левым концом
- Чтобы найти ответ, нужно для каждого отрезка проверить 3 кандидата на минимальное пересечение.

«D. Разрезание торта»

- Идея задачи — Михненко Алексей, ВШЭ
- Разработка задачи — Михненко Алексей, ВШЭ

Постановка задачи

- Дана таблица $n \times m$, некоторые клетки которой покрашены.
- Надо проверить можно ли разрезать эту таблицу горизонтальными или вертикальными разрезами так, чтобы в каждой отрезанной части была ровно одна покрашенная клетка.

Идеи

- Если все покрашенные клетки находятся в разных строках или столбцах, то можно провести только горизонтальные или только вертикальные разрезы.
- Если сделано a горизонтальных разрезов и b вертикальных, то количество отрезанных кусков будет $(a+1)(b+1)$, поэтому $(a+1)(b+1) = k$.

Идеи

- Пусть a — количество горизонтальных разрезов. Тогда первый вертикальный разрез должен проходить между первыми $a+1$ и вторыми $a+1$ покрашенными клетками (в порядке возрастания столбцов, в котором они находятся). Соответственно второй разрез будет проходить между вторыми $a+1$ и третьими $a+1$ клетками и так далее.
- Аналогичные рассуждения можно проделать, если зафиксировать количество вертикальных разрезов.

Решение

- Пусть мы зафиксировали количество горизонтальных и вертикальных разрезов, тогда мы знаем, в каких местах нам надо провести разрезы.
- Таким образом, надо только проверить, что такие разрезы действительно корректны.
- Проверить корректность нетрудно, так как для каждой покрашенной клетки мы определили в какой части она будет находиться, поэтому надо лишь проверить, что все части различны.

Решение

- Пусть D — количество делителей числа k. Тогда, так как $(a+1)(b+1)=k$, количество интересных пар для a и b будет ровно D.
- Если заранее отсортировать покрашенные клетки в порядке возрастания строк и столбцов, построение разрезов и проверку их на корректность можно реализовать за $\mathcal{O}(k)$.
- Таким образом, такое решение работает за $\mathcal{O}(Dk)$, что достаточно быстро, так как при таких ограничениях $D \leqslant 128$.

«E. Печатная машинка»

- Идея задачи — Поликарпов Андрей, ВШЭ
- Разработка задачи — Поликарпов Андрей, ВШЭ

Постановка задачи

- Есть машинка из n ячеек с числами.
- Также есть каретка, которая перемещается вправо. Операция перемещения из n в 1 называется сбросом.
- Каретка может нести в себе одно число, и тем самым изменять содержимое ячеек.
- Есть запросы сдвига всех чисел в ячейках, а также разворота порядка ячеек.
- Требуется сказать, сколько минимально нужно сбросов каретки, чтобы все числа лежали в своих ячейках.

Без запросов

- Давайте решим задачу, если нет запросов
- Ключевым объектом для нас будут такие ячейки, что число в них меньше номера ячейки. Заметим, что за один сброс каретки, мы можем перенести не более одного такого числа. Таким образом у нас есть нижняя оценка на ответ.

Без запросов

- Давайте построим граф с ребрами $i \to a[i]$. Тогда он распадется на циклы.
- Давайте найдем первую позицию, где $a[i] \neq i$, возьмем число из этой позиции, переложим его в $a[i]$, возьмем число из позиции $a[i]$, и так далее.
- Тогда мы поставим весь цикл на свои места. Сколько будет сбросов каретки? Ровно столько, сколько есть ребер, таких что $i < a[i]$.
- Таким образом в точности равен количеству позиций, где $i < a[i]$.

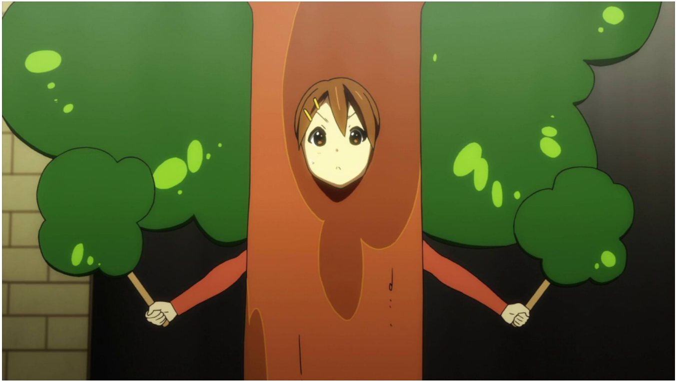
А теперь запросы!

- Давайте научимся решать для сдвигов.
- Найдем для каждой ячейки такие позиции начала массива, что он в этой конфигурации будет удовлетворять $i < a[i]$.
- Нетрудно заметить, что для конкретной ячейки такие индексы будут образовывать отрезок (возможно префикс + суффикс).
- Давайте заведем отдельный массив и будем прибавлять на этих отрезках один. Тогда в i-й позиции будет ответ, если массив начинается с i ячейки.

И еще запросы!

- Давайте научимся решать для переворота массива.
- Нетрудно заметить, что для этого можно просто поддерживать всю предыдущую структуру, но с изначальной перевернутой конфигурацией ячеек.
- Таким образом итоговое решение работает за $\mathcal{O}(n+q)$.

«F. Дерево на Манхеттене»



- Идея задачи — Новиков Владимир, ВШЭ
- Разработка задачи — Маркелов Игорь, ВШЭ

Постановка задачи

- Дано корневое дерево на n вершинах.
- Предком вершины v является вершина с номером p_v, причем на ребре соединяющем данные вершины записано число c_v.
- Нужно сопоставить каждой вершине число x_v так, чтобы $(x_1, \ldots, x_n)$ образовывали перестановку чисел от 1 до n, для каждого поддерева вершины занимали последовательный отрезок значений x_i и сумма $\sum_{v=2}^{n} |x_v - x_{p_v}| * c_v$ была минимальной.

Решения на подгруппы

- Самым простым решениям являются решение за $O(n! * n)$ в котором нам достаточно рассмотреть все перестановки чисел и для каждой перестановки посчитать заданную сумму.
- Для решения последующих подгрупп воспользуемся методом динамического программирования.

Решения на подгруппы

- Пусть $sz(v)$ — размер поддерева с корнем в вершине v. Определим $dp_{vertex,position}$ как минимальную стоимость, что мы можем сопоставить x_i для вершин поддерева значения от 1 до $sz(v)$ так, что $\sum_{v \in subtree(v), v \neq vertex}^{n} |x_v - x_{p_v}| * c_v$ минимальна и вершине v сопоставлено значение $position$. Т.е $x_v = position$. Иными словами мы "выкладываем"поддерево на отрезок так, чтобы мы заплатили наименьшую стоимость за все ребра в этом поддереве, не учитывая на данный момент стоимость из вершины $vertex$ в ее родителя.

Решения на подгруппы

- Такую динамику мы можем пересчитывать, перебирая в текущем корне (вершине *vertex*) то, в каком порядке мы выложим поддеревья детей нашей текущей вершины. Такие решения проходят 2 либо 3 подгруппы в зависимости от реализации.

Решения на подгруппы

- Улучшим предыдущее решение. Для этого сначала заметим, что нам необязательно фиксировать позицию корня в поддереве в нашем динамическом программировании. Вместо $dp_{vertex,position}$ будем хранить состояния dp_{vertex}, но в отличие от предыдущего подхода мы будем учитывать расстояние от корня поддерева до ближайшего края подотрезка (помноженное на стоимость ребра в родителя).

Решения на подгруппы

- Теперь когда мы будем фиксировать порядок подотрезков в поддереве нам достаточно будет посчитать суммарную длину блоков, которые отделяют наш корень от родителя.
- Остается понять, какие блоки выгоднее ставить ближе к корню поддерева. Можно заметить, что достаточно добавлять блоки в порядке возрастания $\frac{c_v}{sz(v)}$. Чтобы доказать этот факт можно явно посчитать, когда выгодно менять местами два соседних блока (и получить нужную формулу).

Решение на полный балл

- Теперь мы знаем, что мы можем отсортировать блоки в порядке возрастания $\frac{c_v}{sz(v)}$. Остается разобраться какие блоки пойдут влево, а какие блоки пойдут вправо от текущего корня.
- Для этого запустим вспомогательное дп, которое напоминает рюкзак. После рассмотрения первых i поддеревьев будем хранить минимальную стоимость, которую мы могли получить, если поддеревья на суммарный размер k попали левее корня.

Решение на полный балл

- Такое решение работает за $O(n^2)$ и набирает полный балл. Для доказательства асимптотики нужно заметить, что вспомогательное дп работает за $O(subtrees * n)$, что в сумме дает $O(n^2)$.