Всероссийская олимпиада им. Келдыша

Разбор задач

Москва, Сочи, 16 июня 2019 г.

А. «Чунга-Чанга»

- Идея задачи — Григорий Резников
- Разработка задачи — Андрей Чулков

Постановка задачи

- У Маши есть x чижиков, а у Саши есть y чижиков. Сначала какая-то одна девочка даст другой сколько-то денег, затем они идут в магазин и каждая купит на все деньги кокосов по z чижиков за кокос.
- Они хотят максимизировать суммарное число полученных кокосов, а при равенстве минимизировать число денег которое одна другой передаст.

Решение за $\mathcal{O}(max(x, y))$ (50+ баллов)

- Переберём сколько одна девочка передаёт другой чижиков.
- Обновляем ответ как $\langle max \lfloor \frac{x-\delta}{z} \rfloor + \lfloor \frac{y+\delta}{z} \rfloor, min\, \delta \rangle$
- Симметрично для передачи денег в другую сторону.

Полное решение

- Несложно вычислить сколько мы купим кокосов: $k = \lfloor \frac{x+y}{z} \rfloor$.
- Если $k = \lfloor \frac{x}{z} \rfloor + \lfloor \frac{y}{z} \rfloor$, то ответ $\langle k, 0 \rangle$.
- Заметим, что нет смысла передавать $\geqslant z$ чижиков, потому что передающий мог бы сам купить кокосов на целую часть переданных z.
- Следует передать ровно столько чижиков, чтобы у принимающей стороны стал нулевой остаток денег на z.
- $\langle k, \min(z - (x \bmod z), z - (y \bmod z)) \rangle$.

B. «Разделение числа»

- Идея задачи — Михаил Тихомиров
- Разработка задачи — Михаил Анопренко

Постановка задачи

- Дано число в десятичной системе счисления.
- Разделить его на две непустые части без ведущих нулей, чтобы минимизировать сумму этих частей.

Решение на 20+ баллов

- Число не содержит нулей и помещается в тип «int».
- Переберем, сколько цифр будет содержать вторая часть. Пусть она содержит c цифр.
- Чтобы получить значение первой части, достаточно взять целую часть от деления n на 10^c.
- Чтобы получить значение второй части, достаточно взять остаток от деления n на 10^c.
- Берём минимум по всем вариантам.

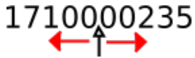
Решение за $\mathcal{O}(len^2)$ на 60+ баллов

- Берём предыдущее решение, но используем длинную арифметику: храним число как последовательность цифр. Сложение и сравнение таких чисел можно делать за длину числа $\mathcal{O}(len)$.
- В языках Python и Java есть встроенная длинная арифметика.
- Добавляем дополнительную проверку, что вторая часть не начинается с нуля; получаем дополнительные баллы.

Ключевая идея и 68+ баллов

- Предположим пока, что нулей нет.
- Заметим, что оптимально разбивать число на две как можно более равные по длине части.
- В частности, для чётной длины $2k$ достаточно попробовать только длину $k + k$.
- А для нечётной длины $2k + 1$ достаточно попробовать варианты $k + (k + 1)$ и $(k + 1) + k$.
- Можно проверять чуть больше вариантов около середины, корректность таких решений доказать немного проще.
- Перебираем $\mathcal{O}(1)$ вариантов, решение за $\mathcal{O}(len)$.

Полное решение



- Если в середине есть нули, то нужно отступить влево и вправо до того момента пока не найдём вариант без лидирующих нулей.
- Как и в прошлом пункте, достаточно проверить лишь по одному варианту в обе стороны, но можно проверить и немного больше.
- Решение за $\mathcal{O}(len)$.

C. «Флаг»

- Идея задачи — Николай Будин
- Разработка задачи — Николай Будин

Постановка задачи

- Дано клетчатое поле, каждая клетка покрашена в какой-то цвет.
- Нужно посчитать количество подпрямоугольников, являющихся “флагами”.
- Прямоугольник является “флагом”, если состоит из трех одноцветных прямоугольников одинаковой высоты, расположенных друг над другом.
- Цвета соседних прямоугольников должны различаться.

Решение за $O(n^6)$ (20+ баллов)

- Переберём подпрямоугольник, высота которого делится на 3
- Разделим его на три трети по высоте, и выберем произвольную клетку в каждой из третей (например, верхние левые)
- Проверим, что цвета всех клеток в трети совпадают с цветом выбранной клетки в этой трети
- Проверим, что цвета соседних третей различаются
- Если все условия выполнены, прибавим к ответу 1

Решение за $O(n^2)$

- Рассмотрим вертикальный отрезок подряд идущих клеток одного цвета
- Он может быть частью средней полосы флага только в том случае, если сверху и снизу от него клетки имеют другой цвет

Решение за $O(n^2)$

- Разобьем все поле на максимальные по включению вертикальные одноцветные отрезки
- Пусть длина отрезка равна l, тогда проверим, что l клеток сверху от него имеют один цвет, и l клеток снизу от него имеют один цвет
- Если так, то запомним значение $(row, col, l, c_1, c_2, c_3)$, где row и col — координаты верхней клетки отрезка, c_1, c_2 и c_3 — цвета клеток над отрезком, клеток отрезка и клеток под отрезком, соответственно

Решение за $O(n^2)$

- Любой флаг образуется множеством значений $(row, lastcol - i, l, c_1, c_2, c_3)$, где $0 \leqslant i < w$, w — ширина флага
- Зафиксируем row, будем перебрать col.
- Для каждого col не сложно определить максимальную длину флага до этого. Для этого надо сравнить (c_1, c_2, c_3) и длину l с тем, которая была в в $col - 1$ и или продлить максимальную ширину на один, либо начать новую.
- Если полученная ширина имеет размер w, то к ответу нужно прибавить w.

D. «Ирригация».

- Идея задачи — Елена Андреева
- Разработка задачи — Егор Чунаев

Постановка задачи

- Каждый год олимпиада проводится в одном из m городов.
- Первые n лет проведение олимиады не поддавалось никакому правилу.
- С $n+1$ года ввели правило, что олимпиада проводится в городе, который проводил её меньше всего раз, а среди таких городов в городе с меньшим номером.
- По информации о первых n годах надо ответить на q запросов о том, в каком городе будет проводится олимпиада в k_i год.

Решение на 20 баллов

- Будет отвечать на каждый запрос по-отдельности.
- Для этого будем вычислять историю проведений по одному году.
- Для этого будем поддерживать текущие кратности городов в списке и дописывать в конец город с наименьшим числом проведений (среди них минимальный по номеру).
- Получили решение за $\mathcal{O}(mk)$ на запрос, то есть за $\mathcal{O}(qkm)$.

Решение на 20+10 баллов

- Заметим, что за первые nm лет все страны проведут одинаковое количество олимпиад.
- Поэтому при $k > nm$ ответ просто равен $(k-1) \pmod{m} + 1$.
- Отвечаем на каждый запрос или за $\mathcal{O}(km)$ или за $\mathcal{O}(1)$. Итого $\mathcal{O}(nm^2)$ на запрос и $O(qm^2n)$ в сумме.

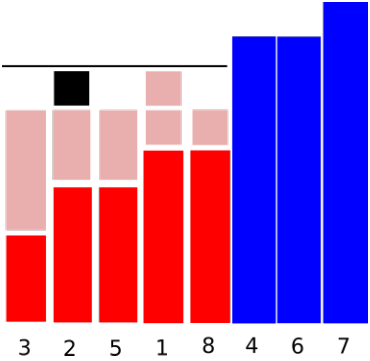
Решение на 40 (+10) баллов

- В тестах суммарной стоимостью 40 баллов $k \leqslant 250\,000$.
- Заведём структуру данных, которая поддерживает все города в отсортированном порядке по количеству проведений, а в случае равенства по номеру города.
- Например можно было воспользоваться $std :: set$, хранящий пары из количества и номера города.

Решение на 40 (+10) баллов

- Имея такую структуру в момент времени cur, мы можем легко перевести её в $cur + 1$.
- Для этого извлечем из структуры минимальный элемент, увеличим количество проведений и добавим элемент обратно.
- Получаем решение за $O(qk \log m)$, которое набирает 40 баллов.
- Реализовав также идею для маленьких тестов с $k > nm$ получится решение, которое набирает 50 баллов.

Полное решение



Полное решение

- Отсортируем страны по количеству проведений соревнования в первые n лет (см картинку).
- Как будет менятся эта диаграмма? Ячейки закрашиваются от нижних строк к верхним, внутри одной строки сортировка происходит по номеру страны.

Полное решение

- Будем построчно закрашивать эту таблицу снизу вверх.
- Когда при закрашивании очередной строки закрашенных клеток становится больше, чем k, то сортируем все не закрашенные клетки текущей строки по номеру.
- $k - S$ в порядке сортировки город и будет ответом, где S — общее количество закрашенных клеток.

Полное решение

- Когда мы «залили» всю диаграму, то дальше каждая строка будет состоять из всех элементов. Тогда ответом является остаток k по модулю m.
- Переходы между строчками делать за $O(1)$ просто поддерживая количество пустых элементов.
- Поскольку до полного «залития» диаграмы мы рассмотрим не более n строк решение работает за $O(q(n+m))$.

E. История одной страны

- Идея задачи — Романов Владимир
- Разработка задачи — Романов Владимир, Алексей Кулдошин

Постановка задачи

На плоскости выбраны n непересекающихся прямоугольников $(x_1, x_2, y_1, y2)$.

Назовем прямоугольную область *правильной*, если ее граница не пересекает прямоугольники из набора, а также выполнено одно из условий:

- Внутри области ровно один прямоугольник
- Область можно разделить на 2 правильные области

Требуется определить, существует ли правильная область, которая содержит все n прямоугольников.

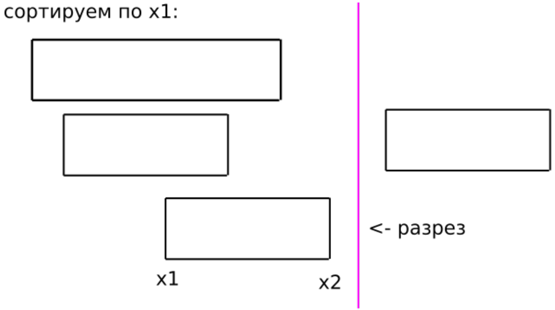
Решение за $O(n^2 \log n)$ (50 баллов)

- Заметим, что если существует разрез, который не пересекает ни один прямоугольник, то его следует провести. Чем меньше прямоугольников в области тем не сложнее разрезать плоскость дальше, значит это нам ничего не испортит.
- Попытаемся найти вертикальный разрез.
- Если не получилось, поищем горизонтальный по аналогии.
- Если разреза нет, то «No».

Решение за $O(n^2 \log n)$ (50 баллов)

- Отсортируем прямоугольники по самой левой точке (по x_1).
- Заметим, что если существует вертикальный разрез, то префикс прямоугольников будут слева от него, а суффикс — справа.
- Критерий «хорошести» разреза: $\max x_2$ из префикса $\leqslant \min x_1$ из суффикса.
- Если находим разрез, запускаемся от двух половин. Время работы: $T(n) = T(x) + T(n-x) + \mathcal{O}(n \log n)$.
- Худший случай $x = 1$, время работы $\mathcal{O}(n^2 log)$.

Критерий хорошести разреза



max x2 слева $\leqslant$ x1 следующего прямоугольника

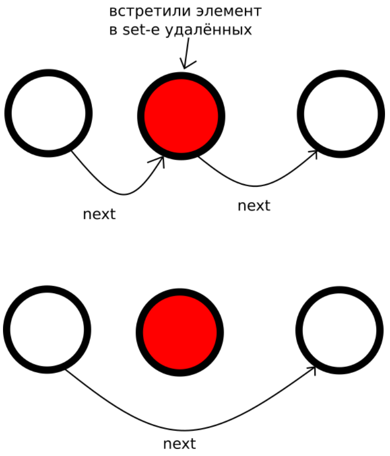
Решение за $O(n \log^2 n)$ (75-100 баллов)

- Ключевая идея: давайте отрезать «меньшее от большего».
- Предположим мы за $\mathcal{O}(1)$ волшебным образом нашли разрез (число x).
- Тогда можно было бы за $\mathcal{O}(x)$ отрезать меньшую часть и запустится от неё рекурсивно. А с большей частью продолжить процесс отрезания.
- Такое решение бы работало за $\mathcal{O}(n \log n)$: при рекурсивном вызове размер задачи уменьшается хотя бы в два раза, а значит глубина логарифмична.
- Осталось разобраться с «магией» — например, для того чтобы реализовать всё вышеперечисленное подошло бы дерево отрезков.

Решение за $O(n \log^2 n)$ (75-100 баллов)

- Давайте отсортируем прямоугольники сортировками по всем 4 возможным направлениям и запустим поиск одновременно по всем 4 направлениям.
- Когда в одном из направлений видим разрез, выкидываем все эти прямоугольники в рекурсивный вызов.
- Внутри нашего рекурсивного вызова помечаем эти прямоугольники как удалённые и просто запускаем проход заново.
- Когда встречаем удалённый прямоугольник пропускаем его. Например это можно делать с помощью связного списка:

Связный список



Решение за $\mathcal{O}(n \log)$ (75-100 баллов)

- Откуда у нас log^2? Один логарифм из-за отрезания меньшего от большего, второй из-за сортировки.
- От логарифма из-за сортировки можно избавится, если аккуратно протащить сортировки внутрь рекурсивного вызова.
- Но это не требовалось.