Задача A. Минимальное покрытие

Посмотрим, в чём заключается задача. Дан большой прямоугольник размера $w_0 \times h_0$, нужно посчитать, сколько ячеек прямоугольной сетки имеют с ним ненулевое пересечение. Сетка задана размерами ячейки $w \times h$ и координатами (x, y) нижнего левого угла какой-то из ячеек.

Полный перебор ячеек неэффективен по времени. Для эффективного решения достаточно найти формулу, вычисляющую ответ в общем случае. Рассмотрим прямые, содержащие стороны известной ячейки сетки. Они разбивают плоскость на 9 частей:

1. Нижняя левая имеет размер $x \times y$
2. Нижняя средняя имеет размер $w \times y$
3. Нижняя правая имеет размер $(w_0 - (x + w)) \times y$
4. Средняя левая имеет размер $x \times h$
5. Центральная нас не интересует, т.к. в ней уже лежит лист кровли
6. Средняя правая имеет размер $(w_0 - (x + w)) \times h$
7. Верхняя левая имеет размер $x \times (h_0 - (y + h))$
8. Верхняя средняя имеет размер $w \times (h_0 - (y + h))$
9. Верхняя правая имеет размер $(w_0 - (x + w)) \times (h_0 - (y + h))$

Зная размер части, количество ячеек в ней можно вычислить формулой: если размер части $a \times b$, то она будет пересекаться с $\lceil \frac{a}{w} \rceil \cdot \lceil \frac{b}{h} \rceil$ ячеек сетки. Применяем формулу по разу к каждой части, кроме центральной, складываем результаты и получаем ответ.

Итоговая асимптотика решения $O(t)$.

Задача B. Подставные палиндромы

Пусть число x имеет вид $\overline{abc}$, то есть, равняется $100 \cdot a + 10 \cdot b + c$. Коли $a = c$, это и так палиндром, ответ 1. Если $c > a$, то $x = \overline{aba} + (c - a)$, ответ 2. В случае $c < a$ и $b \neq 0$ имеем $x = \overline{a(b-1)a} + (10 - a + c)$, снова 2. (Раз $c - a$ и $10 - a + c$ цифры, они и палиндромы).

Остался вариант, когда $c < a$ и $b = 0$. При $c < a - 1$ $(a > 1)$ нам подойдёт $x = \overline{(a-1)9(a-1)} + (11 - a + c)$. И нас интересуют только числа вида $\overline{a0(a-1)}$.

Но заметим, что при $a > 2$ их можно представить, как $\overline{(a-2)9(a-2)} + 111$, а для $a = 1$ подойдёт $99 + 1$. Единственное неразобранное число теперь — 201.

А теперь сюрприз — оно непредставимо в виде суммы 2 палиндромов! Минимальный трёхзначный палиндром — 101, максимальный подходящий — 191, значит, это не могут быть 2 трёхзначных палиндрома или трёхзначный+однозначный (ведь $101 + 101 > 201 > 191 + 9$). Но тогда 201 должно быть равно $\overline{aa} + \overline{1b1}$ при $a \geqslant 1$, что невозможно, ведь последняя цифра суммы будет $a + 1 \neq 1$.

Итого, 201 равно, например, $101 + 99 + 1$, и все случаи разобраны.

Задача C. Два светофора

Посмотрим, в чём заключается задача. Нам дана дробь $\frac{x}{y} \leqslant 1$, и нужно найти наименьшее целое $t \geqslant 0$ такое, что дробь $\frac{x-t}{y-t}$ будет равна дроби $\frac{1}{n}$ для некоторого натурального n (или, что то же самое, $y - t$ будет делиться на $x - t$).

Полный перебор параметра t неэффективен по времени. Однако существует оптимизация перебора, позволяющая решить задачу эффективно.

Положим $d = y - x$. Заметим, что разность между знаменателем и числителем сохраняется с ростом параметра t (то есть d никак не зависит от t). Отсюда разность $y - t$ будет делиться на $x - t$ тогда и только тогда, когда и d будет делиться на $x - t$.

Поскольку мы хотим найти наименьшее подходящее значение t, нам достаточно найти наибольший делитель числа d, не превосходящий x. Это можно сделать за $O(\sqrt{d})$, используя известную технику перебора до корня.

Итоговая асимптотика $O(t \cdot \sqrt{d})$.

Задача D. Торт

Можно посчитать количество частей плоскости по формуле Эйлера: $V - E + F = K + 1$, если считать точки пересечения и концы отрезков вершинами, а части отрезков между ними — рёбрами.

Чтобы посчитать количество вершин и рёбер, нужно для каждого отрезка узнать, со сколькими из оставшихся он пересекается, за $O(n^2)$. Если отрезок i пересекается с k_i отрезками, то:

$$2n + \frac{1}{2}\sum_{i=1}^{n} k_i \text{ — количество вершин (концы отрезков и точки пересечения)}$$

$$\sum_{i=1}^{n}(k_i + 1) \text{ — количество рёбер (на отрезке } i \text{ — } k_i + 1 \text{ ребро)}$$

После этого можно построить граф, в котором вершины — это отрезки и две вершины соединены ребром, если их отрезки пересекаются. Любым обходом этого графа (dfs или bfs) можно узнать количество компонент связности в нём, что равно количеству компонент связности графа, к которому применяется формула Эйлера.

Асимптотика — $O(n^2)$.

Задача E. Самый длинный отрезок

Будем решать задачу методом динамического программирования.

Определим величину: $dp(len, mxlen) =$ Кол-во массивов a длины len, у которых $L(a) = mxlen$, считаем что значение m зафиксировано при подсчёте динамики.

Заметим, что ответ на задачу — это $\frac{1}{m^n} \sum\limits_{mxlen=1}^{n} mxlen \cdot dp(n, mxlen)$.

Базой динамики сделаем $dp(len, len) = m$, массивы состоящие из одного отрезка.

Будем считать динамику «вперёд», то есть к имеющемуся массиву будем добавлять отрезок одинаковых элементов. Заметим, что значение на этом отрезке можно выбрать $(m - 1)$ способом, так как оно не должно совпадать с последним элементом нынешнего массива.

Таким образом, из состояния $dp(len, mxlen)$ нужно прибавить $(m-1) \cdot dp(len, mxlen)$ в состояния вида $dp(len + cur, \max(cur, mxlen))$.

Разобьём прибавления на два вида.

- к $dp(len + cur, mxlen)$, где $1 \leqslant cur \leqslant mxlen$.
- к $dp(len + cur, cur)$, где $mxlen < cur$.

Заметим, что первое прибавление происходит на кусочке столбца динамики, а второе на кусочке диагонали.

Для эффективной реализации таких прибавлений, можно использовать массивы префиксных сумм для столбцов и диагоналей.

Задача F. Диагональное чтение

Назовём скобочным балансом строки число открывающихся скобок минус число закрывающихся.

Предположим сначала, что скобочный баланс всей подстроки t равен 0. Тогда она ПСП $\Longleftrightarrow$ баланс на каждом префиксе не отрицателен.

Возьмём нашу последовательность действий — сдвинуть/добавить «(»/добавить «)», которая превращает t в ПСП, и заметим, что можно переместить все операции сдвигов в начало (поменяв индексы, куда мы добавляем «(» и «)»), при этом её длина не увеличится.

Посмотрим на суффикс последовательности сдвигов (они все в начале) из k символов. Если в нём m открывающихся скобок и $k - m$ закрывающихся (очевидно, в оптимальном решении $m \leqslant k - m$), то применение этих k сдвигов можно заменить на дописывание в начало $(k-m)-m$ открывающихся скобок и в конец $(k - m) - m$ закрывающихся (чтобы суммарный баланс остался 0).

Но тогда в случае $2 \cdot (k - 2m) < k \Longleftrightarrow k < 4m$ противоречие с оптимальностью решения.

Если $k = 4m$ (то есть, открывающихся скобок в 3 раза меньше, чем закрывающихся), можно использовать оба подхода. И заметим, что вся эта начальная последовательность сдвигов разбивается на подотрезки 2 видов — из одного символа «)» и с $k = 4m$, начинающихся на «(» (можно проверить, что из дискретной непрерывности мы не перескочим момент, когда k станет меньше $4m$).

А также нам выгодно применять очередной подотрезок сдвигов («)» или с $k = 4m$), если он не выходит за пределы $[l, r]$.

Если же скобочный баланс строки не равен 0, пусть он b. Тогда нам в любом случае придётся добавить к строке $|b|$ скобок, которые отрегулируют его в 0. Коли $b > 0$, нам придётся добавить их закрывающимися (выгодно в конец), что никак не изменит предыдущее решение, просто добавит к нему b действий.

А когда $b < 0$, обозначим минимум балансов всех префиксов за min. Применение отрезков сдвигов увеличивает его, и возьмём отрезок, во время которого тот стал $> b$.

Заметим, что теперь при замене этого сдвига на добавление открывающихся/закрывающихся скобок нам нужно уже не $2 \cdot (k - 2m)$, а меньше действий, ведь часть открывающихся скобок нам не нужно компенсировать (так как их на $|b|$ меньше), наоборот добавляя их мы убиваем 2 зайцев 1 выстрелом.

То есть, и сдвиги с этого момента использовать не выгодно.

Итого, задача свелась к тому, чтобы уметь искать для индекса ближайший справа, что на подотрезке между ними $k = 4m$ (или он просто «)»). Потом делать сдвиги от l, пока эти отрезки не вылезают за r и баланс на конце $\geqslant min - b$ (при $b < 0$). А оставшееся несоответствие ПСП исправить добавлениями «(» и «)».

Давайте предподсчитаем балансы и ближайшие такие справа для каждого индекса (для этого говорим, что «(» добавляет 3, а «)» вычитает 1, назовём это неравномерным балансом, и наша задача — находить следующий префикс с таким же неравномерным балансом, как и у текущего).

А на предподсчитанных подотрезках (скачках на 1) можно насчитать бинарные подъёмы (скачки на 2^k), с помощью которых получится находить место прекращения сдвигов уже за $\log(n)$.

И осталось искать min баланс на подотрезке, для чего построим ДО, получаем $O(\log(n))$ на запрос, $O((n + k)\log(n))$ суммарно.

Задача G. Сессия

Будем называть вариантом экзамена сам экзамен вместе со временем его проведения (основным или досрочным). Таким образом, есть $2n$ вариантов экзаменов.

Для начала опишем, как выяснить, может ли Тихон сдать все экзамены. Данную задачу можно свести к задаче 2SAT: экзамену с номером i соответствует булева переменная a_i, равная 1, если экзамен сдается в основное время, и 0 — если в досрочное. Дизъюнкты соответствуют недопустимым парам вариантов экзаменов. Например, дизъюнкт $(a_i \vee \bar{a_j})$ присутствует в задаче 2SAT, если невозможно сдать и экзамен i досрочно, и экзамен j в основное время. Всего таких дизъюнктов $\mathcal{O}(n^2)$, поэтому нельзя явно их хранить и строить по ним граф импликаций.

Основа решения задачи 2SAT — это обход в глубину графа импликаций. Для эффективности его работы нужно при переборе соседей быстро находить следующую непосещенную вершину. Пусть варианты экзаменов упорядочены по неубыванию дня проведения. Положим prv_i — такой наибольший номер $j < i$, что после сдачи варианта экзамена j Тихон успеет подготовиться к к экзамену i. С помощью дерева отрезков для Range Minimum Query, построенном на prv, и операции «на отреке $[l, r]$ найти самый левый элемент, со значением больше либо равным x», можно эффективно перебирать непосещенных соседей вершины в обходе в глубину.

Перейдем к решению основной задачи. Если с помощью алгоритма выше удалось выяснить, что Тихон не может сдать все экзамены, то следует вывести «-1». Иначе, будем идти от большей даты экзаменов к меньшей, каждый раз пытаться ставить экзамен на досрочное время, когда это не удается сделать, вывести ответ. Пусть V' — множество вершин, соответствующее досрочным вариантам тех экзаменов, которые мы уже поставили на досрочное время. Положим V — множество всех достижимых вершин из V' в графе импликаций. Тогда если в V, содержится и переменная, и ее отрицание, это означает, что текущий экзамен нельзя поставить на досрочное время. Иначе, расстановка, где все переменные из V равны единице, существует: достаточно взять существующее решение, полученное на первом этапе, и изменить в нем все переменные из V. Полученное решение также будет корректно, так как, в противном случае, существует ребро $u \to \bar{v}$, где u равна единице в новом решении и не лежит в V, а v лежит в V; тогда, по свойству графа импликаций, $v \to \bar{u}$, что противоречит тому, что u не лежит в V. Итоговое время работы: $\mathcal{O}(n \log n)$