Разбор задачи «Подарок Диппера»

Составим граф из всех букв и проведем ребра между разными буквами минимальной ценой замены. Посчитаем минимальное расстояние в графе изменений алгоритмом Флойда-Уоршелла за $O(a^3)$, где a — размер алфавита. Расстояние между вершиной a и b в графе соответствует минимальному количеству монет, которое нобходимо, чтоб получить из символа a символ b.

Рассмотрим делители n, только такие числа являются кандидатами на k-строку из n символов. Таких подходящих k будет порядка $O(\sqrt[3]{n})$.

Будем решать для каждого k отдельно. Рассмотрим букву на позициях i, для каждого $i \, mod \, k$ посчитаем количество букв стоящих на таких позициях. Решаем независимо для каждого остатка от деления на k. Для этого переберем букву, которая будет стоять на этих позициях и посчитаем суммарную цену замены на необходимую букву.

Разбор задачи «Шкаф для обуви»

Выпишем условия для пары обуви размера $size$ стоящей на полки с высотой h_i:

$\frac{height}{k} \leqslant h_i \leqslant height \Leftrightarrow h_i \leqslant height \leqslant h_i \cdot k$

$\frac{height}{m_1} \leqslant size \leqslant \frac{height}{m_2}$

Подставляя неравенство на $height$, получаем, что должны выполняться два условия:

$\frac{h_i}{m_1} \leqslant size \Leftrightarrow h_i \leqslant size \cdot m_1$

$size \leqslant \frac{h_i \cdot k}{m_2} \Leftrightarrow size \cdot m_2 \leqslant h_i \cdot k$

Проверяя эти два условия независимо для каждой пары обуви, находим ответ.

Разбор задачи «Цифровая загадка»

Для начала заметим, что наиболее выгодно заменять цифры на 9. При этом, логично, что не нужно заменять 9-ки.

Дальше заметим, что чем больше разряд, тем выгоднее его заменить, так например, в числе 85 выгоднее заменить 8-ку, чем 5-ку.

Еще один полезный факт — понять, что в одинаковых разрядах выгоднее заменять меньшую цифру. Данный факт можно было понять из первого примера в условии.

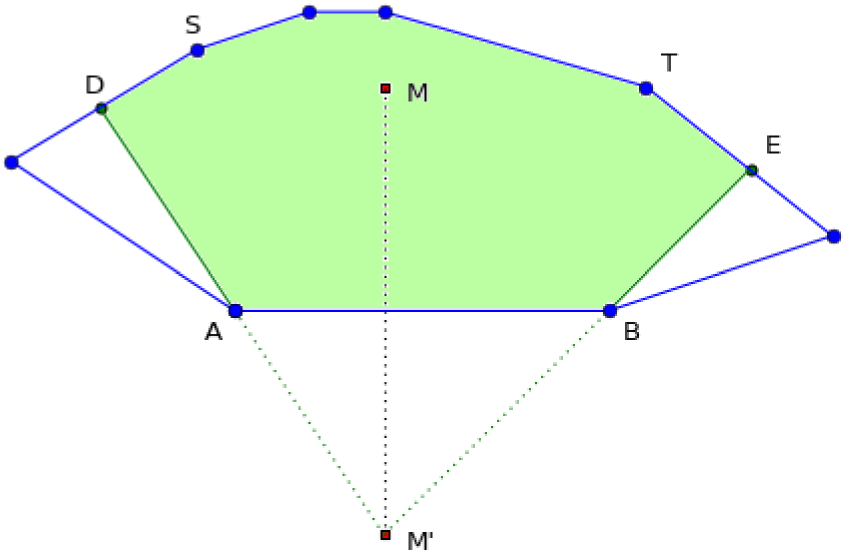
Чтобы решить задачу, нужно преобразовать число в сумму разрядных слагаемых. Для примера $123 = 100 + 20 + 3 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$. Положим все такие слагаемые в массив, задавая их парой из цифры и степени 10-ки: $(1, 2), (2, 1), (3, 0)$.

И наконец, отсортируем этот массив, сначала по уменьшению степеней 10-ки, а потом по увеличению цифры, и выбрать первые k элементов. Для каждой пары (x, y) к ответу прибавим $(9-x)*10^y$.

P. S. Можно сразу положить в массив $(9 - x) * 10^y$ и сортировать уже такие числа.

Разбор задачи «Зеркало»

Будем находить ответ для каждой стороны отдельно.



Пусть M — рабочее место Стэна, AB - сторона-зеркало. Отразим M относительно прямой AB.

Проведём из получившейся точки M' лучи $M'A$ и $M'B$. Требуется вычислить площадь пересечения угла $AM'B$ и многоугльника.

Найдём точки D и E — точки повторного пересечения лучей $M'A$ и $M'B$ с границей многоугольника. Для этого бинарным поиском найдём стороны, с которыми пересекается лучи, и найдём точки пересечения.

Затем найдём площадь полученного многоугольника (на рисунке закрашен зелёным). Его стороны — DA, AB, BE, два куска сторон исходного многоугольника (DS и ET) и некоторая непрерывная последовательность сторон исходного многоугольника (от T до S). Чтобы вычислить его площадь, нужно просуммировать ориентированные площади треугольников, образованных каждой из сторон и началом координат. Посчитаем эти площади для сторон DS, DA, AB, BE и ET, а чтобы найти сумму для сторон от T до S, заранее посчитаем частичные суммы на префиксах.

Время работы решения $O(n \log n)$, так как для каждой из n сторон делается два бинарных поиска.

Разбор задачи «В поисках неизведанного»

Маршруты, о которых говорится в данной задаче — гамильтоновы пути. То есть пути, которые проходят через каждую вершину данного графа ровно по одному разу. Заметим, что если путь w подходит, то «развернутый» путь $\widetilde{w}$ (вершины в котором следуют в обратном порядке) — тоже подходит, так как пути считаются различными, если последовательности вершин в них неодинаковы.

Тогда общее количество таких путей всегда четно. Кроме случая когда $n = 1$, тут ответ, очевидно, равен 1.

Разбор задачи «Тайная комната»

Преобразуем неравенство из условия: $a_i + i < a_j + j$. Теперь нужно найти такое максимальное количество элементов массива, что для каждой пары элементов выполняется следующее: сумма значения и индекса одного элемента меньше, чем сумма значения и индекса другого.

Прибавим к каждому элементу массива его индекс и получим новый массив b. В полученном массиве нужно найти такое максимальное количество элементов, что в каждой паре значение одного элемента строго меньше значения другого, то есть необходимо просто посчитать длину наибольшей возрастающей последовательности в массиве b.

Разбор задачи «Починка хижины»

Для решения этой задачи можно было заметить, что среди чисел вида $\frac{a}{i}$ не больше, чем $2\sqrt{a}$ различных. Можно перебрать значение $\frac{a}{i}$, получить отрезок подходящих значений i, и прибавить значение суммы при данных i к ответу, не забыв, что $\frac{b}{i}$ тоже может принимать разные значения.

Разбор задачи «Очередь к аттракциону»

Несложно заметить, что Диппер должен войти в игру перед первым сдвигом. При этом, если количество колонн нечетное, он должен войти после последнего человека, который войдет в игру до сдвига, а если четное, то сразу же.

Разбор задачи «Диппер и аппарат»

Будем решать задачу в оффлайн. По m запросам создадим события:
для первого типа запросов l r s:

- события начала отрезка $(l,\ 1,\ j)$, где j — номер запроса
- событие конца отрезка $(r,\ 3,\ j)$, где j — номер запроса

для второго типа запросов i x y:

- $(i,\ 2,\ j)$ — событие запрос, где j — номер запроса

Отсортируем события сначала по первой координате, а при равенстве — по второй. При событии 1-го типа — добавляем пару $(j,\ |s|)$ в декартово дерево, при событии 3-го типа — удаляем пару $(j,\ |s|)$ из декартова дерева. В качестве x значения используем j, а $|s|$ используем для того, чтобы поддерживать сумму длин добавленных строк.

При событии 2-го типа нужно выбрать такие строки, которые попали в подстроку-запрос. Для этого выберем в декартовом дереве такой наименьший префикс, что сумма длин строк на нем больше либо равна y из запроса, а также наибольший префикс, что сумма длин строк на нем меньше либо равна x. Это стандартная задача и решается просто спуском по декартову дереву, подобно тому, как делается *split*. Обойдем данное поддерево и сконкатенируем строки в нем. Так как суммарная длина запрашиваемых подстрок не превосходит 10^6, суммарное количество вершин, которые мы обойдем, не превзойдет 10^6.

Итоговая сложность получается $O(m \log m + \sum_1^m s_j)$