Задача A. Похожие подарки

Рассмотрим длину максимального совпадающего префикса s и t; эту длину можно найти за $O(min(|s|, |t|))$. Аналогично находим длину максимального совпадающего суффикса. Предположим, что строка t представляется как конкатенация префикса и суффикса строки s; тогда длина префикса не больше максимального, так же и длина суффикса. Так как нам подойдет любая комбинация размеров, сумма которых равна |t|, то осталось лишь найти длину пересечения найденных ранее суффикса и префикса в строке t. Эта длина равна $|suf|+|pref|-|t|$. Так как мы можем использовать любое(возможно ноль) число символов из пересечения как окончание префикса, а остальные — как начало суффикса, то ответом на задачу будет $|suf|+|pref|-|t|+1$. Решение работает за $O(min(|s|, |t|)$ (не учитывая считывания строк, разумеется).

Задача B. Дом, который построил Джек

Без ограничения общности скажем, что $X \leqslant Y \leqslant Z$.

Поймём, что если длины сторон вдоль oX, oY и oZ равны x, y и z соответственно, то мы сможем поместить в дом максимум $\lfloor \frac{X}{x} \rfloor \cdot \lfloor \frac{Y}{y} \rfloor \cdot \lfloor \frac{Z}{z} \rfloor$ квартир — по $\lfloor \frac{X}{x} \rfloor$, $\lfloor \frac{Y}{y} \rfloor$ и $\lfloor \frac{Z}{z} \rfloor$ вдоль осей независимо.

Заметим, что для любого $1 \leqslant x \leqslant X$ выполнено либо $x \leqslant \sqrt{X}$, либо $\lfloor \frac{X}{x} \rfloor \leqslant \sqrt{X}$. То есть, либо длина стороны квартиры вдоль oX не больше $\sqrt{X}$, либо кол-во квартир вдоль oX не больше $\sqrt{X}$.

Также заметим, что для квартиры со стороной x нам оптимально взять вдоль oX их $\lfloor \frac{X}{x} \rfloor$ штук и наоборот, если вдоль оси oX у нас есть x квартир, длину стороны каждой лучше всего сделать $\lfloor \frac{X}{x} \rfloor$.

Переберём эти значения (длину квартиры или их кол-во вдоль данной оси) для X и Y. Если на данном шаге текущее кол-во квартир получилось t, значит, вдоль Z должно быть ещё не меньше (и, очевидно, оптимально) $\lceil \frac{C}{t} \rceil$ квартир, откуда получаем оптимальное z и вычисляем текущий объём.

Взяв в конце максимум по всем вариантам, получим ответ. А работать всё будет за $\sqrt{X} \cdot \sqrt{Y} = \sqrt{X \cdot Y} = \sqrt{\frac{X \cdot Y \cdot Z}{Z}} = \frac{\sqrt{X \cdot Y \cdot Z}}{\sqrt{Z}} \leqslant \frac{\sqrt{X \cdot Y \cdot Z}}{\sqrt{\sqrt[3]{X \cdot Y \cdot Z}}}$ (так как $Z \to$ максимальная сторона) $= \sqrt[3]{X \cdot Y \cdot Z} \leqslant \sqrt[3]{10^{18}} = 10^6$ итераций перебора, как каждая итерация за $O(1)$.

Задача C. Максимальный разрез

Для начала подвесим дерево за произвольную вершину. Заметим, что при разрезе дерево разбивается на поддерево какой-то вершины и наддерево какого-то ребра.

Заметим, что вершинный диаметр больше рёберного ровно на один, поэтому считать будем рёберный.

Диаметр для каждого поддерева считается динамикой по поддеревьям. Достаточно поддерживать две величины:

- $depth_v$:=глубина поддерева вершины v
- $diam_v$:=диаметр поддерева вершины v

Формулы для пересчёта:

$$C(v) := \text{Множество детей вершины } v \text{ в подвешенном дереве}$$

$$depth_v = 1 + \max_{u \in C(v)} depth_u$$

$$diam_v = \max\{\max_{u \in C(v)} diam_u, \max_{u_1 \neq u_2 \in C(v)} (depth_{u_1} + depth_{u_2} + 2), \max_{u \in (v)} (1 + depth_u)\}$$

Для пересчёта удобно найти две наибольших глубины среди детей.

Вычисление диаметра наддерева происходит по тем же формулам, но сверху вниз. Сделаем обход в глубину, который как параметры принимает также глубину и диаметр наддерева.

Заметим, что при спуске в ребёнка нам нельзя учитывать глубину этого ребёнка. Давайте сначала сохраним все глубины (считая наддерево). То есть у нас есть какое-то множество глубин, из которого мы должны временно удалить число, взять два максимума и вернуть обратно. Это можно сделать, если поддерживать три максимума вместо двух. Аналогично, достаточно хранить два наибольших диаметра.

Итоговая асимптотика решения $O(n)$.

Задача D. Расшифруй строку

Для начала посмотрим, в чём заключается задача. Нам даны хеши определённых подстрок закольцованной строки, и нужно восстановить эту строку по имеющимся хешам.

Заметим, что имея хеш h строки и хеш h_1 префикса этой строки, можно легко вычислить хеш h_2 оставшегося суффикса этой строки как $h_2 = h - h_1 \cdot p^{|h|-|h_1|}$ (естественно, производя все операции по модулю M). Данное преобразование и является ключом к решению.

По имеющимся двум наборам хешей подстрок длин a и b (полагая $a > b$) будем последовательно строить новый набор хешей для подстрок длины $a - b$, затем заменим хеши длины a на хеши длины $a-b$ и повторим. Будем продолжать до тех пор, пока не получим хеши подстрок длины 1, по которым напрямую восстановим строку s. То, что такая цепочка вычитаний сведётся к единице, следует из взаимопростоты исходных чисел a и b (числа преобразуются аналогично алгоритму Евклида).

Решение в описанном виде имеет асимптотику $O(n^2)$: $O(n)$ вычитаний, при каждом из которых за $O(n)$ вычисляется новый набор хешей. Это недостаточно эффективно в ограничениях задачи. Однако, количество шагов в алгоритме Евклида можно сократить с $O(n)$ до $O(\log n)$, используя взятие остатка вместо вычитания (т.е. $a \bmod b$ вместо $a - b$). Для этого только требуется вычитать на каждом шаге хеш, являющийся конкатенацией набора из нескольких имеющихся хешей, идущих через b друг от друга. Такое вычитание можно реализовать за $O(n)$, используя метод двух указателей.

Итоговая асимптотика решения $O(n \log n)$.

Задача E. Дальние враги

Чтобы найти две самые дальние точки из двух разных множеств, сначала необходимо найти выпуклые оболочки этих двух множеств, так как искомые точки находятся на этих выпуклых оболочках. Докажем это. Допустим, есть две точки из двух разных множеств — A (из первого множества) и B (из второго множества), расстояние между которыми самое большое (рис. 5). Тогда можно выбрать такие точки D и F на двух разных выпуклых оболочках, что $DF \geqslant AB$. Чтобы найти D и F, сначала найдём точку пересечения луча BA с выпуклой оболочкой первого множества — C (если их две, то дальнюю от B). Тогда $CB \geqslant AB$. Заметим, что C лежит на какой-то стороне выпуклой оболочки первого множества. Тогда пусть D — один из концов этой стороны, причём такой что $\angle ACD \geqslant 90°$. Тогда $DB \geqslant CB \geqslant AB$. Аналогично найдём вершину второй выпуклой оболочки — F, такую что $DF \geqslant DB \geqslant AB$.

Таким образом, задача свелась к поиску двух самых удалённых точек на двух разных выпуклых многоугольниках. Чтобы найти две такие точки, сначала отразим второй многоугольник симметрично относительно начала координат (то есть если у точки были координаты $(x,\ y)$, то стали $(-x,\ -y)$). После этого найдём сумму Минковского первого многоугольника и второго отражённого. Теперь самая удалённая от начала координат вершина многоугольника суммы Минковского — это результат «вычитания» точки второго многоугольника из точки первого многоугольника, причём эти две точки — искомые самые удалённые. Это верно, так как для любой внутренней точки из суммы Минковского $(x,\ y)$ верно, что $x = x_1 - x_2$ и $y = y_1 - y_2$ для некоторой внутренней точки $(x_1,\ y_1)$ первого многоугольника и некоторой внутренней точки $(x_2,\ y_2)$ второго многоугольника. Значит расстояние от точки $(x,\ y)$ до $(0,\ 0)$ — это расстояние между точками $(x_1,\ y_1)$ и $(x_2,\ y_2)$, так как $\sqrt{x^2+y^2} = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$. Так как сумма Минковского содержит всевозможные «разности» точек первого и второго многоугольников, то она также содержит и разность искомых самых удалённых точек. Также искомая точка суммы Минковского (самая удалённая от начала координат) — это вершина многоугольника суммы Минковского, а значит достаточно для каждой вершины запомнить, из каких точек выпуклых оболочек она получилась, чтобы при нахождении самой удалённой от начала координат можно было узнать ответ.

Асимптотика — $O(n \log n + m \log m)$, так как построение выпуклых оболочек — $O(n \log n) + O(m \log m)$, нахождение суммы Минковского — $O(n + m)$.

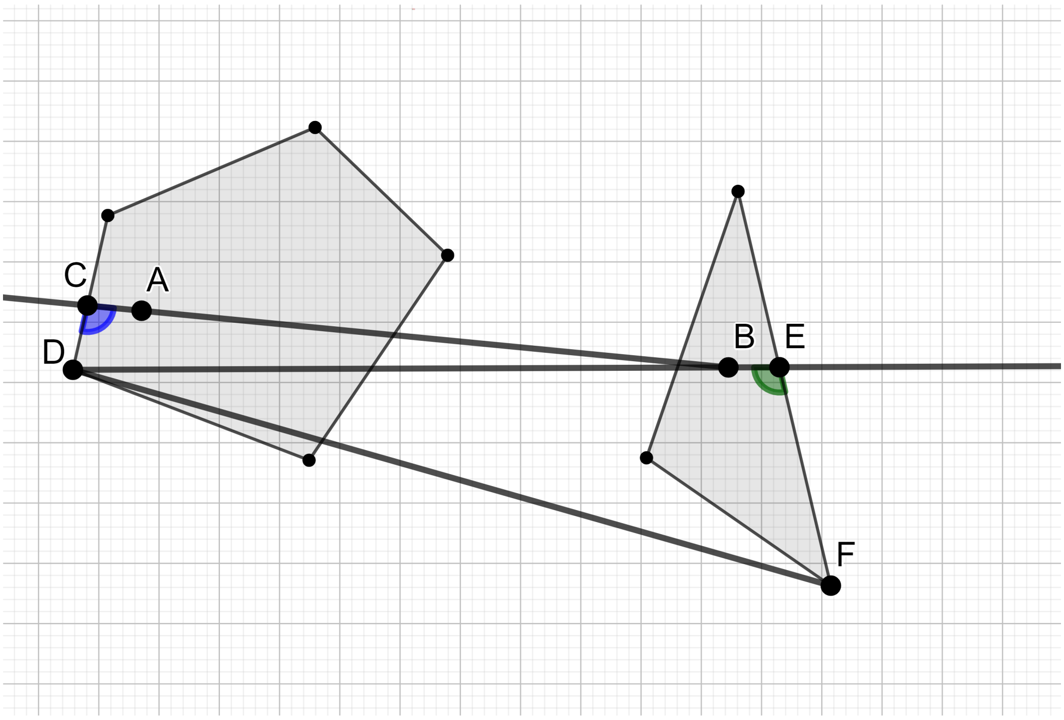

Рис. 1: Две самые дальние точки на выпуклых оболочках