Заключительный этап VIII открытой олимпиады по программированию

Разбор задач второго дня

Маленькая сказка о фиолетовом бобре

В некотором царстве, в некотором государстве жил веселый фиолетовый бобер. Однажды пошел он на качелях качаться. Качался - качался, и вдруг качели сломались. Упал бобер на землю и горько заплакал. И с горя стал он тут яму копать. Вдруг видит, что что-то в яме шевелится. Посмотрел повнимательней, а это личинка майского жука. "Ну", - подумал бобер: "-Женюсь!!! А вдруг это заколдованная принцесса?!" Женился фиолетовый бобер на принцессе, и хотя это оказалась вовсе не принцесса, иногда они были счастливы.

Борисов Алексей (11 лет), 5 "А" класс, Гимназия N5.

Автор идеи: Глеб Евстропов
Разработчик: Владислав Епифанов
Автор разбора: Михаил Пядеркин

Формальная постановка

Дан массив из чисел -2, -1, 0, 1, 2. Необходимо найти отрезок с максимальным произведением.

Простейшее решение

Переберем границы отрезка [l, r].

Для каждого из них в типе long long вычислим произведение элементов на этом отрезке.

$O(N^3)$, 30 баллов

Простейшее решение

Как хранить произведение?

Оно либо равно нулю, либо имеет вид $+2^n$ или -2^n, поэтому достаточно хранить знак и показатель степени n.

Будем перебирать l. Затем будем перебирать r от l до n, пересчитывая произведение по ходу, домножая его на a_r.

$O(N^2)$ и 60 баллов

Избавляемся от нулей

Нули разбивают массив на несколько частей, каждую часть можно обработать отдельно.

Ответ для части отрицателен только в случае, если часть состоит из одного отрицательного числа.

Правильное решение

Будем идти r слева направо и поддерживать “частичные произведения на префиксах” $s_r=a_1*a_2*\ldots*a_r$. Заметим, что произведение чисел на отрезке [l, r] равно s_r / s_{l-1}. Таким образом, при фиксированной правой границе нас интересует минимальное s_{l-1} **такого же знака,** так как мы ищем положительный ответ.

Поддерживаем минимальное по модулю каждого знака.

O(N) и 100 баллов

08:57 **«Пиксели» не торжествуют: шедевр утрачен**

Дерзкое ограбление совершено в Парижском музее современного искусства. Похищено множество экспонатов, наиболее известный из которых — картина «Пиксели торжествуют» киберкубиста Этьена Бурсье-Мужено. Представитель Национального архива Франции Армель ле Гофф обратилась с призывом к программистскому сообществу: для восстановления картины требуются квалифицированные специалисты.

Читать полностью

08:59 «Маленькая сказка о фиолетовом бобре»: новый бестселлер знаменитой Веры

09:05 Родители капитулируют, Игорь едет в магазин игрушек

09:08 Чересчур быстрая доставка пиццы поставила под угрозу вечеринку Вениамина

19:04 Волейболист Пётр подозревается в незаконном владении трактором

20:40 Безобидная игра на дереве стала причиной разлада в молодой семье Вовы и Марины

11:08 Таинственная Вера: миф или реальность?

20:10 В исчезновении динозавров обвинили тёмную материю

Автор идеи:
Максим Ахмедов

Автор разбора
и разработчик:
Антон Полднев

Формальная постановка

Даны N фрагментов чёрно-белой картины, некоторые из которых инвертированы. Требуется восстановить исходную картину.

1 группа: $N \leq 10$, $S \leq 1000$, $HW \leq 1000$

Для каждого фрагмента переберём, инвертирован он или нет. Для очередного варианта (всего их 2^N) восстановим пиксели, входящие в фрагменты. Остальные красим в белый цвет.

Сложность — $O(2^N \cdot (HW+S))$, **30 баллов.**

2 группа: $\mathbf{N \le 500, S \le 2 \cdot 10^4, HW \le 5 \cdot 10^5}$

Представим, что мы построили следующий граф: каждому фрагменту соответствует **вершина,** пересекающиеся фрагменты соединены **ребром.**
На ребре стоит **метка** 0, если эти фрагменты либо оба инвертированы, либо оба не инвертированы (одинаковые пиксели имеют одинаковый цвет), и 1, если инвертированности фрагментов различны.

Как он нам поможет?

2 группа: $\mathbf{N \le 500}$, $\boldsymbol{S \le 2 \cdot 10^4}$, $\mathbf{HW \le 5 \cdot 10^5}$

Нужно определить, какие фрагменты инвертированы, а какие нет, то есть раскрасить вершины графа в два цвета 0 и 1 так, чтобы:

- ребро с меткой 0 → одинаковый цвет концов;
- ребро с меткой 1 → разный цвет концов.

Это делается с помощью поиска в глубину аналогично алгоритму проверки графа на двудольность.

2 группа: $\mathbf{N \leq 500}$, $\boldsymbol{S \leq 2 \cdot 10^4}$, $\mathbf{HW \leq 5 \cdot 10^5}$

Мы нашли какую-то картину, а хотим ещё и максимизировать количество нулей. Единственная свобода действий — целиком инвертировать все фрагменты в некоторой компоненте связности графа.

Для этого добавим в DFS подсчёт количества нулей и единиц. Нулей в компоненте оказалось меньше → инвертируем её.

2 группа: $\mathbf{N \leq 500}$, $\boldsymbol{S \leq 2 \cdot 10^4}$, $\mathbf{HW \leq 5 \cdot 10^5}$

Знаем, как использовать граф. Как его строить?

Давайте для каждого пикселя картины в процессе считывания будем хранить количество фрагментов, её покрывающих. Считали фрагмент — прошлись по всем его пикселям и для каждого пикселя по всем покрывающим его фрагментам.

2 группа: $\mathbf{N \leq 500,\ S \leq 2 \cdot 10^4,\ HW \leq 5 \cdot 10^5}$

Метки рёбер можно хранить в матрице смежности. В соответствующем цикле насчитываем эти метки и заодно контролируем непротиворечивость данных.

Сложность — O(NS), **60 баллов.**

Оптимизируем

Можно поступить проще: считали блок → перебрали, какие блоки с ним пересекаются → добавили ребро, а метку определили по одному пикселю из пересечения.

Корректность проконтролируем в самом конце, ничего страшного.

Сложность — $O(N^2)$, **72 балла.**

Полное решение

Граф — это хорошо, но он может быть полным. Пусть вершинами будут не только фрагменты, но и пиксели поля. Проведём ребро (*cell*, *frag*), если фрагмент *frag* содержит пиксель *cell.* Тогда получим N+HW вершин и S рёбер.

Рёбер стало меньше, поскольку фрагменты, ранее соединённые ребром, теперь связаны косвенно через общие пиксели.

Полное решение

Вершины этого графа тоже нужно раскрасить в два цвета: цвет фрагмента — это его инвертированность, цвет пикселя — это цвет пикселя :)

Метка на ребре (*cell*, *frag*) — исходный цвет пикселя *cell* во фрагменте *frag.*

При раскрашивании графа метки учитываются аналогичным образом: метка 1 означает, что концы ребра имеют разные цвета, а метка 0 — одинаковые.

Сложность — O(HW+S+N), **100 баллов.**

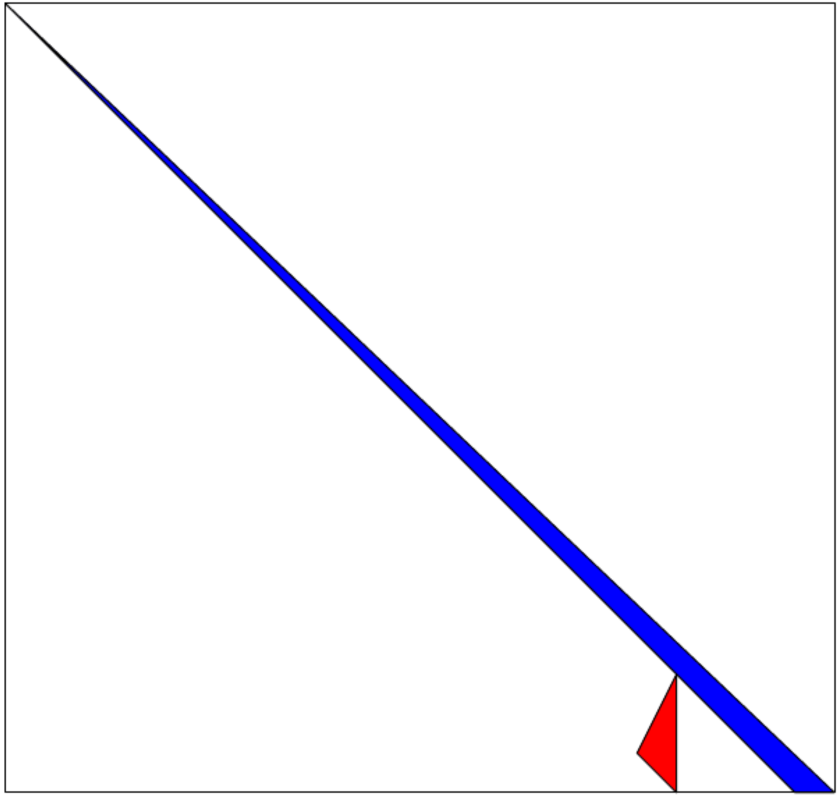
Задача «Игорь и игрушки»

Автор идеи: жюри олимпиады
Разработчик: Роман Андреев
Автор разбора: Роман Андреев

Формальная постановка

Даны N выпуклых многоульников и Q запросов. Каждый запрос состоит в следующем - взять два многоугольника, свдинуть их параллельно оси O_X и сказать, какую минимальную ширину можно получить после сдвига.

Тест №4



Предварительные рассуждения 1

Давайте сначала зафиксируем, какой из двух многоугольников идет первым, а какой вторым. Далее заметим, что если мы сначала раздвинем многоугольники, а затем начнем их сдвигать, то в момент касания найдется вершина одного из многоугольников, которая упрется в сторону другого многоугольника.

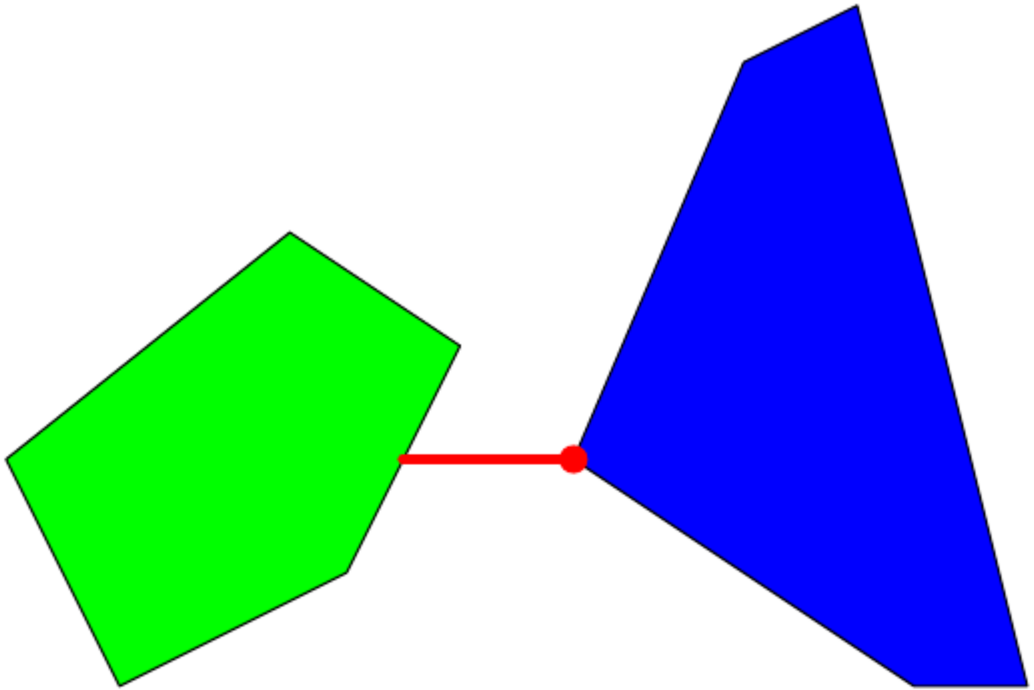
Предварительные рассуждения 2

Чтобы вычислить минимальную ширину можно сложить ширины двух многоугольников и вычесть то расстояние, на которое их надо сдвинуть из состояния, когда коробки придвинуты вплотную.

Решение за $O(QN^2)$

Давайте честно переберем вершину касания в первом многоугольнике и ребро касания во втором многоульнике(и наоборот). Если провести ось параллельную оси O_X через вершину, то она должна пересечь отрезок и мы получим как раз то расстояние, которое нам надо минимизировать.

Поясняющая картинка



Решение за O(QNlogN)

Заметим, что при фиксации вершины первого многоугольника всего одно ребро подходит под наш критерий (два, если наша прямая прошла через вершину, но тогда нам подойдет любое из них). Можно это ребро искать бинарным поиском по значениям координаты O_y вершин левой половины правого многоугольника (или правой половины левого многоугольника соответственно).

Решение за O(QN)

Если мы будем перебирать вершины левого многоугольника в порядке возрастания координат оси O_y, то и ребро правого многоугольника тоже будет двигаться наверх. Тогда можно воспользоваться методом двух указателей и получить нужную асимптотику.

Решение за O(QlogNlogC)

Введем функцию F(h), которая будет вычислять расстояние между точками пересечения правого и левого многоугольника с прямой, параллельной оси O_x на высоте h. Заметим, что эта функция выпуклая. Тогда ее минимум можно искать тернарным поиском. Вычислять значение F (h) можно за O(logN) также, как и в решении за O(QNlogN).

Решение за O(QlogNlogS)

Из того, что касание происходит по вершине, следует, что тернарный поиск можно делать только по координатам вершин. Также заметим, что вместо тернарного поиска по функции можно делать бинарный поиск по ее производной (в нашем случае производной будет разность наклонов ребер, которую можно сверять с нулем с помощью векторного

Решение за $O((S + Q)\log S)$

Сейчас нашей задачей будет применить метод двух указателей из решения за $O(QN)$ для быстрого нахождения позиций ребер в бинарных поисках для всех запросов. Воспользуемся методом параллельных бинарных поисков. Давайте рассмотрим все бинарные поиски для всех запросов в совокупности.

O((S + Q)logS) (продолжение)

На первой итерации во всех запросах мы будем искать во всех многоугольниках ребра, соответствующие одной и той же высоте. Их мы можем обработать за линию. На втором этапе запросы поделятся на левые и правые. На каждой итерации будем поддерживать список запросов, отсортированных в порядке возрастания отрезков запросов бинарных поисков. Тогда для каждого многоугольника заведем свой указатель и будем их двигать. В итоге каждая итерация бинарного поиска будет обрабатываться за O(S + Q).

Задача «Пицца для вечеринки»

Автор идеи, разработчик и автор разбора
Глеб Евстропов

Формальная постановка

Имеется N пар (a_i, b_i), выбрать максимальное по размеру подмножество пар P такое, что существует такая перестановка элементов P,

что для любого j выполнено:

$$\sum_{k=1}^{j-1} a_k \leqslant b_j$$

$N \leq 10$, 20 баллов

Перебор всех перестановок всех подмножеств

и проверка: $2^N \cdot N! \cdot N = 1024 \cdot 3628800 \cdot 10 \approx 3.6 \cdot 10^{10}$.

Заметим, что достаточно перебрать все перестановки

и для каждой найти наибольший подходящий префикс.

Итого: $N! \cdot N \leq 4 \cdot 10^6$.

Перестановочный метод

Классическое применение перестановочного метода: сравнение двух соседних элементов ответа.

Пусть (a_i, b_i) и (a_j, b_j) - соседни в P.

$\min(b_i - a_i - a_j, b_j - a_j)$ или $\min(b_j - a_i - a_j, b_i - a_i)$?

Легко убедиться, что i имеет смысл ставить раньше j только если $a_i + b_i \leq a_j + b_j$.

$N \leq 20$, 40 баллов

Отсортируем пары по $a_i + b_i$.

Переберем подмножества и проверим.

Итого: $2^N \cdot N \approx 10^7$

$N \le 5000$, 60 баллов

$f_{i,j}$ - рассмотрели первых i (в порядке сортировки), взяли ровно j, какую минимальную сумму a можно получить?

$f_{i,j} \Rightarrow f_{i+1,j}$ - ничего не меняем.

$f_{i,j} + a_{i+1} \Rightarrow f_{i+1,j+1}$, если $f_{i,j} \le b_{i+1}$ - пытаемся улучшить значение.

Итого: $N^2 = 2.5 \cdot 10^7$

Замечания о свойствах $f_{i,j}$

$f_{i,j} \leq f_{i,j+1}$ - монотонность, очевидно.

$f_{i,j+1} - f_{i,j} \leq f_{i,j+2} - f_{i,j+1}$ - выпуклость вниз, необходимо доказать

Выпуклость легко доказывается по индукции, но слайды слишком малы, чтобы вместить это доказательство.

$N \le 300\ 000$, 100 баллов

Монотонность и выпуклость группируют переходы.

Монотонность ограничивает префикс, $f_{i,j} \le b_{j+1}$

Выпуклость делает выгодным суффикс, $f_{i,j+1} - f_{i,j} \ge a_{j+1}$

Переходы - групповая операция. Требуется прибавление на отрезке и вставка/удаление.

Итого: декартово дерево, $O(N \log N)$

Конструктивное решение

Минусы: реализация, отсутствие сертификата.

Сертификат - персистентное дерево. Не лучший выбор!

Храним текущий ответ, при добавлении новой пары всегда добавляем её в решение. Если не получилось - выбрасываем элемент с максимальным a_i.

Домашнее задание: доказать это решение.