

Заключительный этап IX открытой олимпиады по программированию

Разбор задач второго дня

Задача «Вид из окна»



Автор идеи: Михаил Пядёркин

Разработчик и автор разбора: Иван Лахтанов

Формальная постановка

На плоскости даны несколько ломаных, идущих “слева направо”. Абсциссы начал и концов всех ломаных совпадают.

Необходимо найти любую пару пересекающихся ломаных, если такая существует.

1 группа: $|A|, |B| \leq 50, N \leq 50$, 20 баллов

Для всех пар целых a, b , таких что $A \leq a < b \leq B$, и для всех пар i, j проверяем, правда ли, что в точке с абсциссой a ломаная i находится строго ниже ломаной j , а в точке с абсциссой b строго выше ломаной j .

Если для каждой ломаной предсчитать ее значения в целых точках, то такое решение будет работать за $O((B-A)^2 N^2)$.

2 группа: $N \leq 100$, $M \leq 100$

Замечание: если две ломанные пересекаются, то какое-то звено первой ломаной пересекает некоторое звено второй ломаной.

2 группа: $N \leq 100$, $M \leq 100$

Переберем все пары звеньев, принадлежащих разным ломаным, и с помощью векторного произведения проверим их на пересечение. Сложность этого решения $O(N^2M^2)$ и оно получает 40 баллов.

3 группа: $N \leq 400$, $M \leq 400$

Рассмотрим некоторую пару ломаных и зафиксируем у первой ломаной какое-то звено. Это звено может пересекать лишь те звенья второй ломаной, которые начинаются не позже, чем заканчивается выбранное звено, и заканчиваются не раньше, чем оно начинается.

3 группа: $N \leq 400$, $M \leq 400$

Эта идея подсказывает нам, что пересекаться слишком далекие звенья двух ломаных друг с другом не могут.

Как же найти все пары звеньев, которые могут друг с другом пересекаться? При помощи метода двух указателей!

3 группа: $N \leq 400$, $M \leq 400$

Теперь для каждой пары ломаных при помощи метода двух указателей можно проверить на пересечение все пары интересующих нас звеньев, и это дает нам решение, работающее за $O(N^2M)$ и получающее **60** баллов.

4 группа: $N \leq 100000$, $\Sigma M \leq 200000$

Оказывается, что перебирать все пары ломаных и проверять их на пересечение не нужно.

Достаточно отсортировать все ломаные по “высоте” и проверять на пересечение только соседей в порядке сортировки...

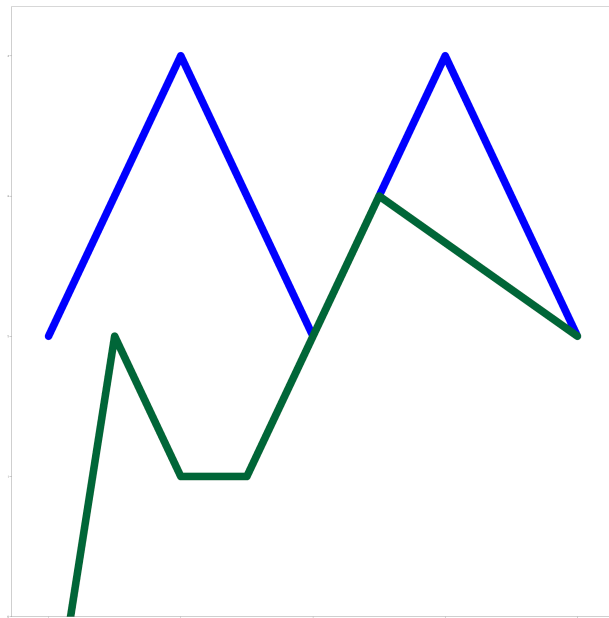
4 группа: $N \leq 100000$, $\Sigma M \leq 200000$

Легко заметить, что даже при непересекающихся ломаных ни сортировка по высоте первой вершины, ни сортировка по максимальной высоте ломаной, не подходит. А что же подходит? Сортировка по площади области, находящейся под ломаной!

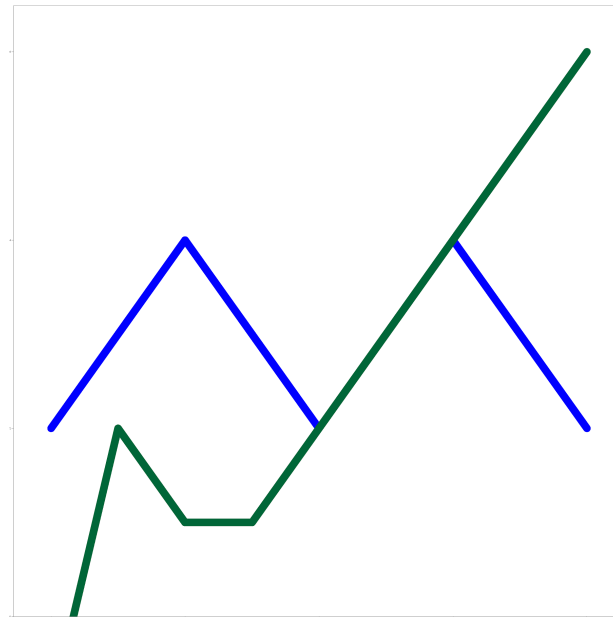
4 группа: $N \leq 100000$, $\Sigma M \leq 200000$

Итак, если отсортировать все ломанные по площадям областей, находящихся под ломаными, а затем всех соседей в порядке сортировки проверить на пересечение при помощи метода двух указателей, то получится правильное решение, работающее за $O(N \log N + M)$ и получающее 100 баллов.

Картинки с интересными тестами



Тест №5



Тест №6

Задача «Простоквашино»



Автор идеи: Михаил Густокашин
Разработчик и автор разбора: Вильям Саакян

Формальная постановка

- Есть N непересекающихся отрезков посещения кладовки.
- Во время посещения свет должен быть включен. После окончания лампочку можно выключить или оставить включенной.
- После K циклов выключения лампочка ломается.
- Одна лампочка стоит C рублей. Минута электроэнергии D рублей.
- Нужно минимизировать потраченные деньги

1 группа: $K = 1$, $N \leq 50$, $a_i, b_i \leq 100$

Так как **$K = 1$** , то после каждого отрезка мы можем либо испортить лампочку и выключить свет, либо оставить её включенной и заплатить деньги за интервал между посещениями.

Для каждого интервала сравниваем цену лампочки и цену интервала, выбирая меньшее значение

Решение за **$O(N)$** . Набирает **20** баллов

2 группа: $K, N \leq 100$

Используем динамическое программирование **dp**
[i][j] - минимальное количество денег, если мы включим лампочку после **i**-ого посещения и это будет её **j**-ое включение.

Изначально все значения **dp** будут равны количеству денег, равному стоимости электричества одной включенной лампочки на протяжении всего времени

2 группа: $K, N \leq 100$

Переберем предпоследнее включение лампочки(q)

Возможны 2 различных перехода:

- $j = 1$

$$dp[i][j] = \min(dp[q][k] - C * (a[i] - b[i - 1]) + D)$$

- $j \neq 1$

$$dp[i][j] = \min(dp[q][j - 1] - C * (a[i] - b[i - 1]))$$

2 группа: $K, N \leq 100$

Состояний $N * K$, в каждом $O(N)$ переходов.

Итого за $O(N^2 * K)$. Набирает 40 баллов

3 группа: $K, N \leq 1000$

Давайте упростим динамику.

Зачем перебирать все состояния, если можно перебрать только предыдущее?

Удобнее теперь прибавлять значения.

3 группа: $K, N \leq 1000$

Опять возможны 2 различных перехода:

- $j = 1$

$$dp[i][j] = \min(dp[i-1][k] + d, dp[i-1][j] + C * (a[i] - b[i-1]))$$

- $j \neq 1$

$$dp[i][j] = \min(dp[i-1][j-1], dp[i-1][j] + C * (a[i] - b[i-1]))$$

3 группа: $K, N \leq 1000$

Состояний $N * K$, в каждом $O(1)$ переходов.

Итого за $O(N * K)$. Набирает 60 баллов

4 группа: $K, N \leq 200,000$

Замечание 1:

Если мы используем ровно одну лампочку, то мы можем выключить ее $K - 1$ раз, то есть не платить за $K - 1$ интервал между посещениями

Если мы используем i лампочек, то мы можем не платить за $i * K - 1$ интервал. Но тогда придется заплатить $(i - 1) * D$ рублей за новые лампочки

4 группа: $K, N \leq 200,000$

Замечание 2:

Не платить логичнее всего за самые дорогие интервалы, т.е. самые длинные



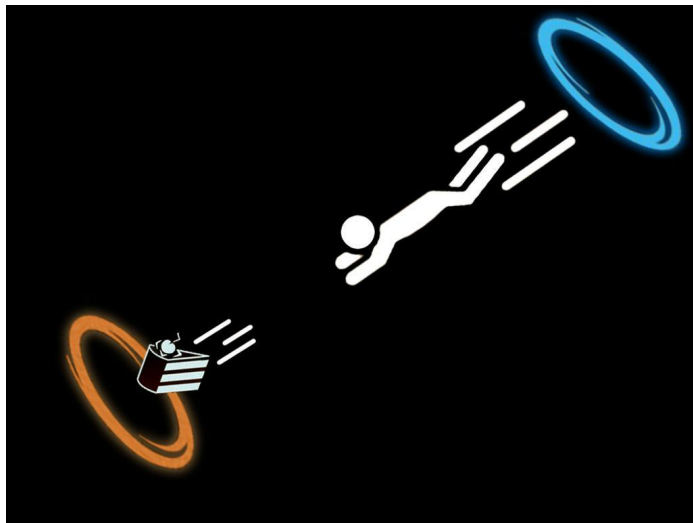
4 группа: $K, N \leq 200,000$

Тогда отсортируем все интервалы между посещениями и, перебрав количество лампочек от **1** до **N / K** , найдем оптимальный вариант.

Тогда надо отсортировать отрезки за **$O(N \log N)$** и пройти по ним за **$O(N)$** .

Итого за **$O(N \log N)$** . Набирает **100** баллов

Задача «The cake is a lie»



Автор идеи, разработчик и автор разбора: Степан Каргальцев

Формальная постановка

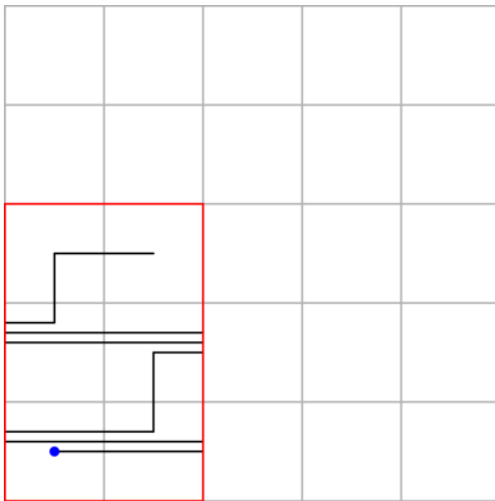
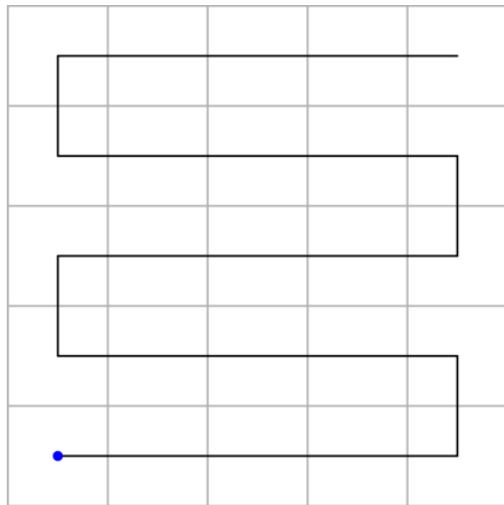
- Ничего не дано.
- Нужно обойти все торы (прямоугольники, описанные в задаче) с площадью, не более заданной (20, 250, 1400 или 6000).
- Все торы площади S нужно обойти не более, чем за $2 \cdot 10^5, 4 \cdot S^2, 4 \cdot S\sqrt{S}, 32 \cdot S$ ходов соответственно (в зависимости от того, сколько групп тестов мы хотим пройти)

Замечание

Так как мы находимся на торе, можно считать, что наше изначальное положение - $(0, 0)$ или любые другие координаты, так как систему координат можно сдвинуть

1 группа: $S \leq 20$, $Q \leq 2 \cdot 10^5$

Давайте обходить квадрат 20×20 как-нибудь,
например, змейкой:



1 группа: $S \leq 20$, $Q \leq 2 \cdot 10^5$

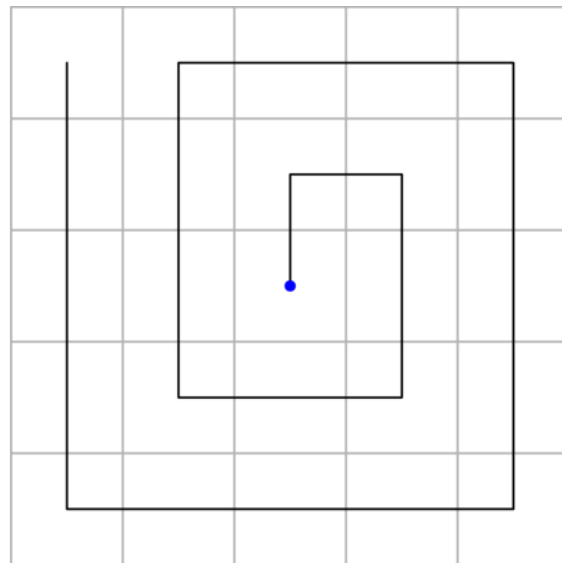
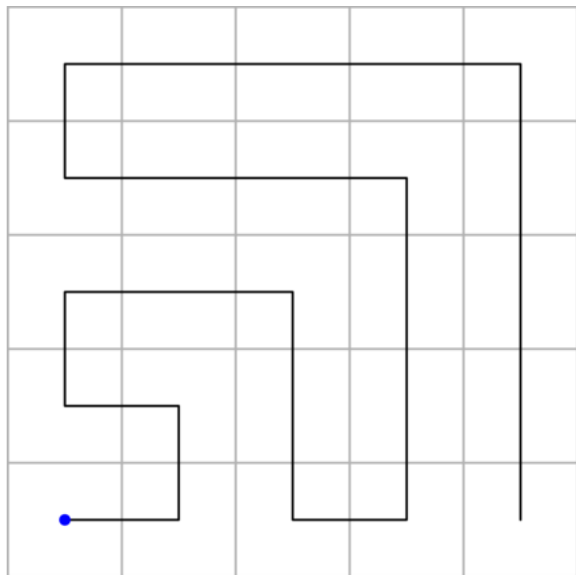
Так как ограничения очень маленькие, каждый раз выбирать новое случайное направление и ходить туда пока не найдем торт тоже получит АС.

2 группа: $S \leq 250$, $Q \leq 4 \times S^2$

Просто обходить квадрат 250×250 как в прошлый раз не получится: если поле было размера 2×2 , то 2×4^2 шагов вправо мы сделаем раньше, чем хотя бы один шаг вверх ($2 \times 4^2 = 32 < 250$).

2 группа: $S \leq 250$, $Q \leq 4 \square S^2$

Но можно обходить хитро!



2 группа: $S \leq 250$, $Q \leq 4 \square S^2$

В первом случае, например, клетку с координатами (x, y) мы посетим не позже, чем через $\max(x^2, y^2)$ шагов.

Кстати, решение, сдвигающееся в случайном направлении на каждом шаге, проходит и эту группу!

Ключевая идея

Пусть у нас есть алгоритм, работающий за $f(S)$,
причем $f(S) / S$ - неубывает. Тогда последовательный
запуск алгоритма при $S = 1, 2, 4, \dots, 2^k$ будет работать
не медленнее, чем за $2 f(2^k)$

Ключевая идея

$$\begin{aligned} & f(1) + f(2) + \dots + f(2^k) = \\ &= 1 \cdot \frac{f(1)}{1} + 2 \cdot \frac{f(1)}{2} + \dots + 2^k \cdot \frac{f(2^k)}{2^k} \leq \\ &\leq 1 \cdot \frac{f(2^k)}{2^k} + 2 \cdot \frac{f(2^k)}{2^k} + \dots + 2^k \cdot \frac{f(2^k)}{2^k} \leq \\ &\leq (1 + 2 + \dots + 2^k) \cdot \frac{f(2^k)}{2^k} \\ &\leq 2 \cdot f(2^k) \end{aligned}$$

3 группа: $S \leq 1400$, $Q \leq 2^{2.5} \cdot S\sqrt{S}$

Пусть нам каким-то магическим образом известно, что площадь находится в пределах от $S/2$ до S

Тогда одна из сторон тора не превосходит $\sqrt{S}$, а другая - S . Тогда, если обойти прямоугольники $S \times \sqrt{S}$ и $\sqrt{S} \times S$ мы заведомо обойдем весь тор (т.е. все торы площади от $S/2$ до S) за $2 \cdot S\sqrt{S}$ шагов

3 группа: $S \leq 1400$, $Q \leq 2^{2.5} \cdot S\sqrt{S}$

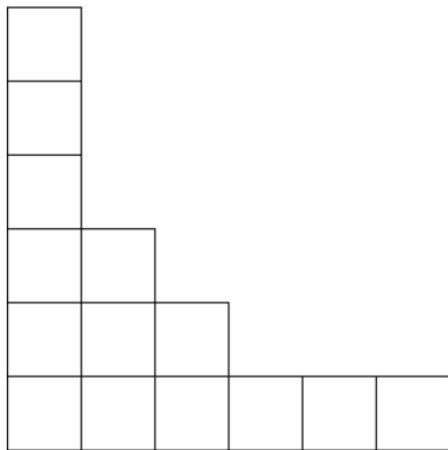
Теперь будем последовательно обходить предыдущим алгоритмом торы площади от 1 до 2, от 2 до 4, от 4 до 8, ..., от $2^{\lceil \log_2 S \rceil - 1}$ до $2^{\lceil \log_2 S \rceil}$

По ключевой идее, это будет работать не медленнее, чем за $2 \cdot 2 \cdot (2S\sqrt{2S}) = 2^{3.5} \cdot S\sqrt{S}$. При более аккуратном анализе получится, что константа меньше.

4 группа: $S \leq 6000$, $Q \leq 32$ $\square$ S

Посмотрим, как выглядит объединение
прямоугольников площади не более S ?

Гипербола!



4 группа: $S \leq 6000$, $Q \leq 32 \square S$

Давайте обойдем эту гиперболу. Сделаем S шагов вверх, шаг вправо, $S / 2$ вверх, шаг вправо, $S / 3$ вверх, шаг вправо, $\dots$, S / S шагов вверх

Всего мы сделаем $S + S / 2 + S / 3 + \dots + S / S =$

$$S \cdot \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{S}\right) \sim S \ln S$$

4 группа: $S \leq 6000$, $Q \leq 32 \square S$

Будем последовательно обходить гиперболы
отвечающие площадям $1, 2, 4, \dots, 2^{\lceil \log_2 S \rceil}$

Итого мы потратим не более чем

$2 \cdot \ln 6000 \cdot S \sim 18S$ шагов и победим

Что еще набирает 100 баллов

Если выбрать 2 направления по разным осям (вверх и вправо, например) и каждый раз ходить случайно в одно из этих двух, то математическое ожидание времени обхода тора $\sim S \log S$. Это даже можно доказать!



Если интересно, здесь можно ознакомиться с соответствующей теорией:
<http://www.cs.elte.hu/~lovasz/erdos.pdf>, Th 2.7

Задача «Урок физкультуры»



Автор идеи: Глеб Евстропов

Разработчик и автор разбора: Владислав Епифанов

Формальная постановка

- Дана перестановка из N элементов.
- Нужно сделать не более одной транспозиции, чтобы минимизировать количество инверсий в получившейся перестановке.

1-3 группы: $N \leq 30$, $N \leq 300$, $N \leq 1000$

Переберем два элемента перестановки, и посчитаем на сколько изменится количество инверсий, если поменять эти два числа местами. На самом деле, нас будут интересовать только числа расположенные между переставляемыми числами, поэтому получится решение за $O(N^3)$ с хорошей константой. Это решение набирает **30** баллов.

4-5 группы: $N \leq 2000$, $N \leq 5000$

Можно попытаться добавить в кубическое решение различные отсечения по ответу и по времени (и перебирать пары в случайном порядке). Такое решение может набрать от **30** до **50** баллов.

4-5 группы: $N \leq 2000$, $N \leq 5000$

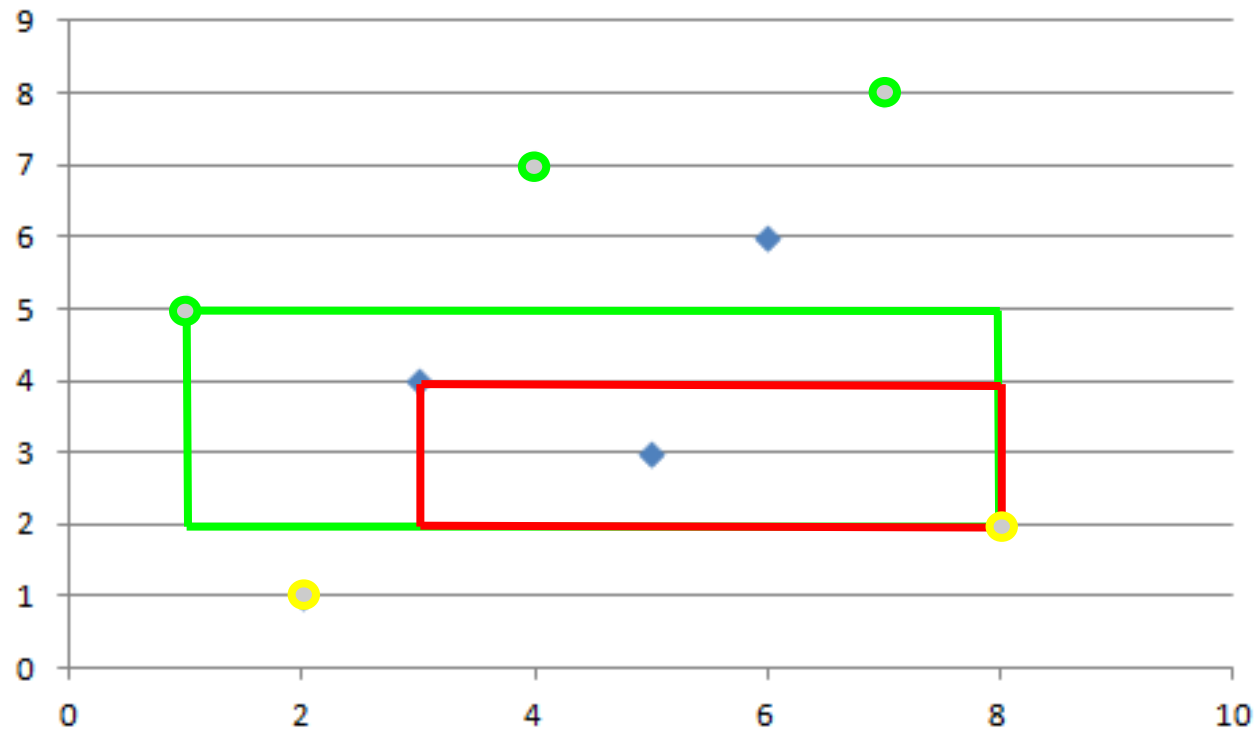
Давайте сохраним в массиве для каждой пары индексов (i, j) количество чисел на отрезке от i до j меньших/больших i -го/ j -го числа. Эти массивы можно заполнить за $O(N^2)$. Используя эти величины, можно теперь для любой пары индексов легко посчитать, как изменится количество инверсий после транспозиции этих элементов. Это решение использует $O(N^2)$ памяти и набирает **50** баллов.

6-10 группы: $N \leq 1000000$

Пусть нам дана перестановка $p_1, \dots, p_n$. Поставим на плоскости точки с координатами (i, p_i) . Будем менять два элемента с номерами $i < j$ такие, что $p_i > p_j$ (если $p_i < p_j$, то количество инверсий только увеличится).

Нетрудно заметить, что после такой транспозиции, количество инверсий уменьшится на $1+2 \cdot (\text{кол-во поставленных точек внутри прямоугольника с вершинами в точках } (i, p_i) \text{ и } (j, p_j))$.

6-10 группы: $N \leq 1000000$

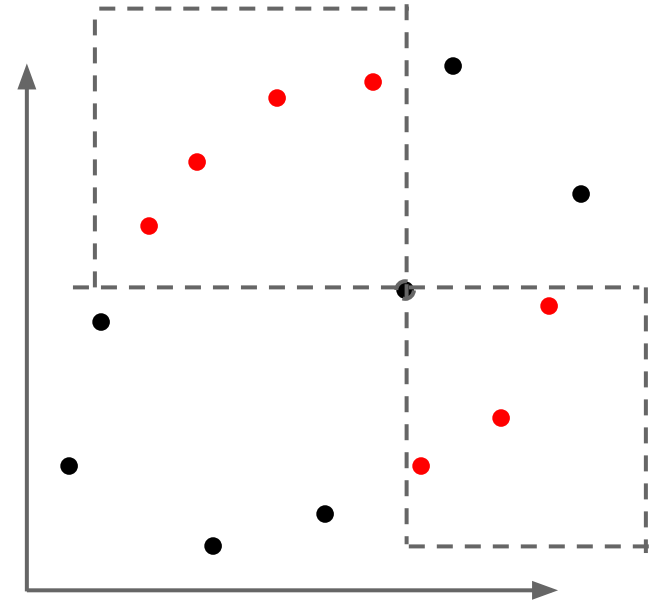


6-10 группы: $N \leq 1000000$

Если для некоторого числа p_i , левее него встречается число большее, чем p_i , то это число не имеет смысла рассматривать в качестве первого числа для обмена (иначе можно рассмотреть то большее число, и ответ только улучшиться). Аналогично, если есть число правее и меньше. Назовем такие числа хорошими левыми/правыми концами.

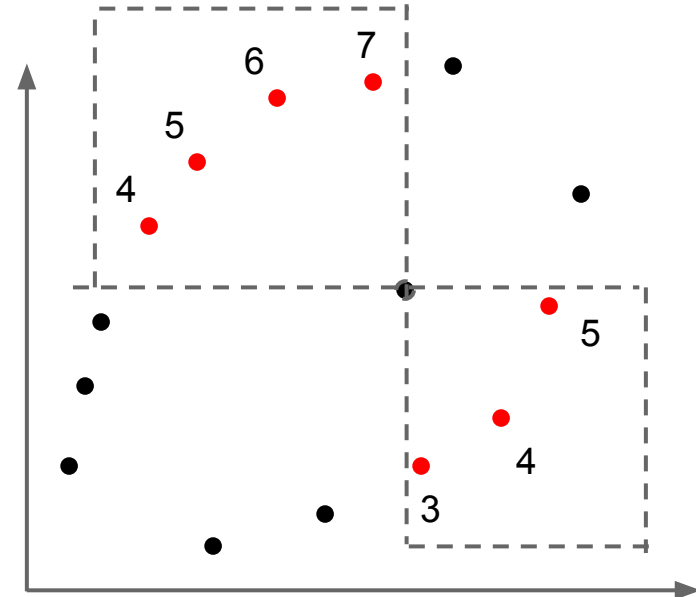
6-10 группы: $N \leq 1000000$

Найдем все хорошие левые и правые концы. Для каждого числа p_i не попавшего ни в одно из этих множеств, определим, в какие прямоугольники оно попадает. Для фиксированного p_i подойдут все хорошие левые концы, точки, соответствующие которым лежат левее и выше точки (i, p_i) .

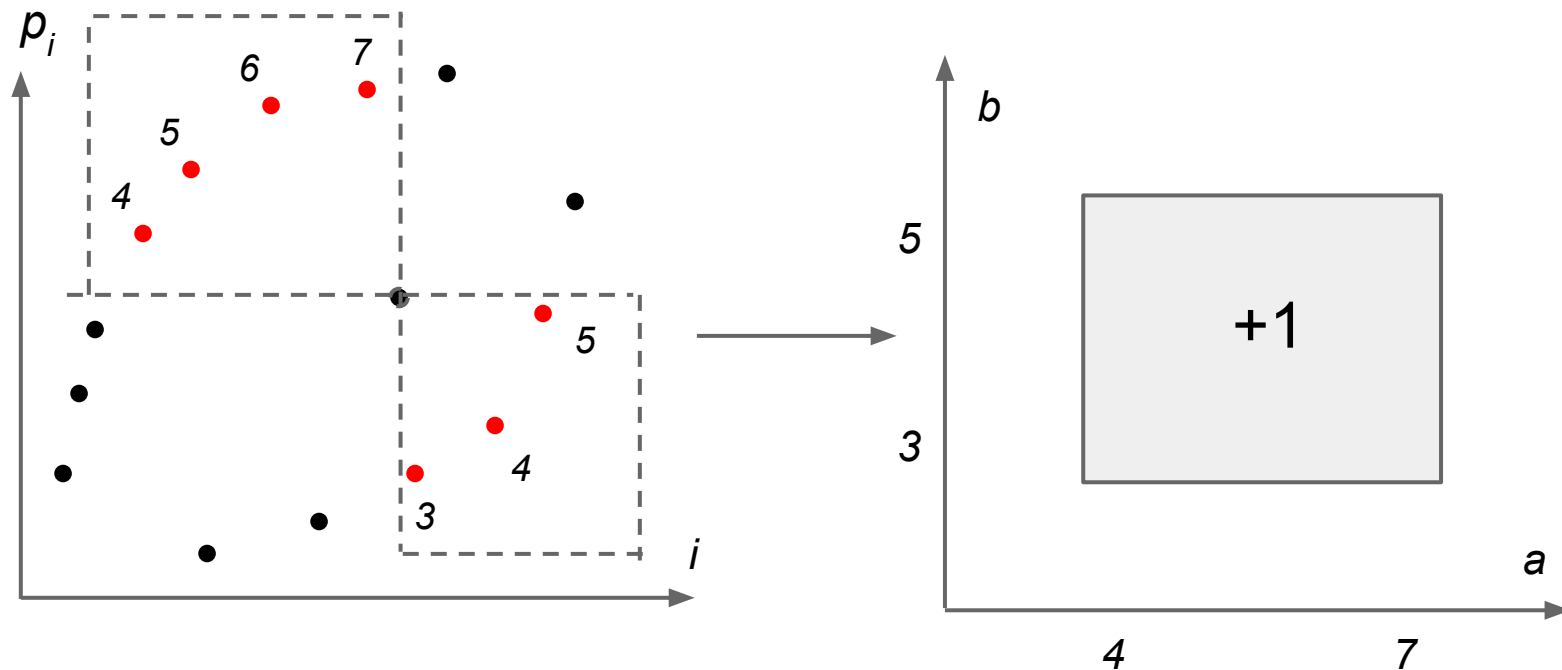


6-10 группы: $N \leq 1000000$

Так как все хорошие левые концы образуют возрастающую последовательность, то подходящие левые концы будут образовывать непрерывный отрезок. Аналогично для правых концов (они должны лежать правее и ниже). Введем теперь систему координат, на осях которой будут написаны хорошие левые и правые концы.



Сведение к задаче на прямоугольниках



6-10 группы: $N \leq 1000000$

Каждому числу в этой системе координат будет соответствовать некоторый прямоугольник. В итоге мы свели задачу к следующей: есть $\leq N$ прямоугольников, нужно найти точку, покрытую максимальным кол-вом прямоугольников. Эту задачу можно решить с помощью метода сканирующей прямой и простой структуры данных (дерево отрезков/фенвика) за время $O(N \log N)$.