

Т-Поколение 2026–2027

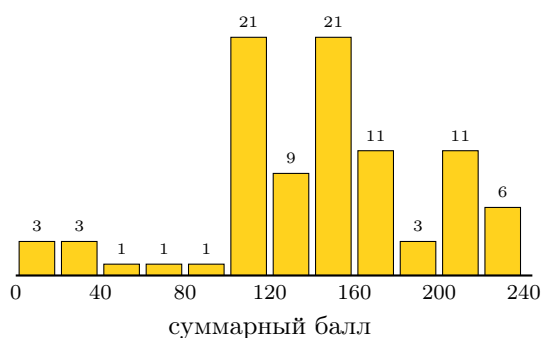
Тренировочный тур №1

Разбор задач

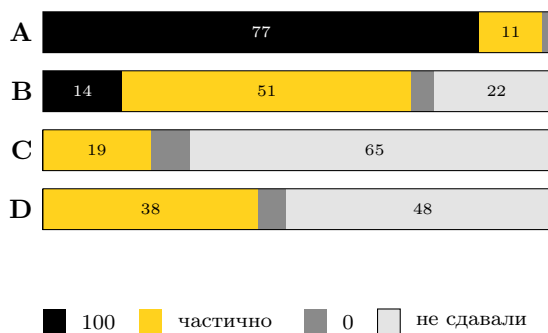
Интересная статистика. На момент написания разбора тур написал 91 человек, наибольший полученный балл оказался равен 238. Смотри только на результаты тура может показаться, что есть решаемые задачи А и В, и нерешаемые гробы С и D. Но, вообще говоря, задачи С и D гробами не являются, а для их решения требуется только понимать вещи уровня "а так ведь тоже можно!". Поэтому советуем внимательно ознакомиться с разбором и учесть эти техники из этих задач на будущее.

В задаче С была подгруппа $m = 1$, которую незаслуженно проигнорировали все участники, хотя как раз таки в ней можно было путем последовательных замечаний получить какое-то решение, пусть и без идеи для решения всей задачи на полный балл.

Распределение результатов



Результаты по задачам



Рядом с заголовком разбора каждой подгруппы указано, сколько баллов можно получить, если решить эту подгруппу. Меньшее число включает в себя только баллы за это подгруппу и те, от которых она зависит, а большее число включает в себя совокупный балл, если решить все ранее разобранные подгруппы.

Задача А. Прогулка по Петербургу

Напомним условие задачи: есть вертикальные дороги $x = x_1 < \dots < x_n$ и горизонтальные $y = y_1 < \dots < y_m$. В каждом из q запросов Кирилл начинает путь на перекрёстке (x_a, y_b) , а далее гуляет по дорогам в направлении увеличения координат (каждую секунду на 1 увеличивается одна из координат). Пусть t — время, прошедшее с начала прогулки (оно же пройденный путь). Между перекрёстками Кирилл просто идёт по своей дороге, а на перекрёстке выбирает направление исходя из чётности t : если t чётно, то он идёт вверх, а если нечётно, то вправо. Требуется определить, где он окажется в момент $t = d$.

На рис. 1 показана первая прогулка из примера. Дороги там раскрашены по чётности координаты: ниже станет ясно, что именно чётности определяют, где Кирилл поворачивает.

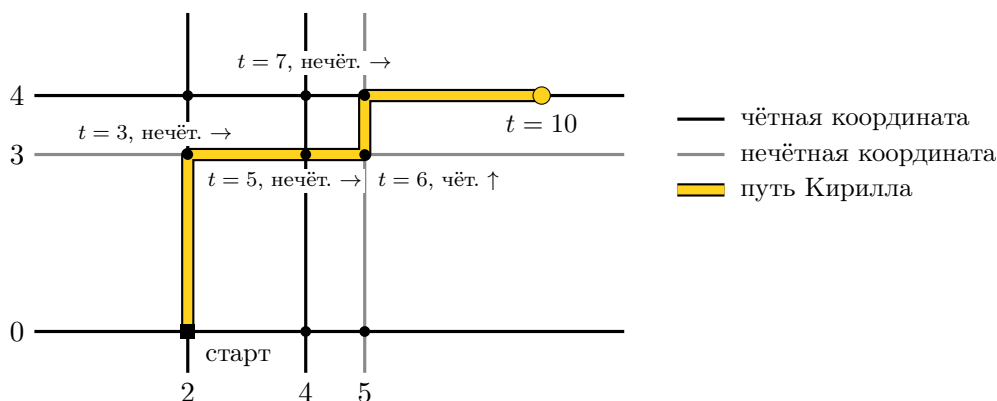


Рис. 1: Первая прогулка из примера: старт в $(2, 0)$, $d = 10$, финиш в $(8, 4)$.

Решение подзадачи 1

10 баллов

В этой подзадаче $d \leq 50$, поэтому можно честно промоделировать прогулку секунда за секундой. Для текущей точки (x, y) легко за $\mathcal{O}(n + m)$ определить, лежит ли она на вертикальной дороге (x есть среди $x_1, \dots, x_n$) и на горизонтальной (y есть среди $y_1, \dots, y_m$). Если только на вертикальной — идём вверх, если только на горизонтальной — вправо. Если на обеих, то точка (x, y) — перекрёсток, и выбор надо делать исходя из чётности t . Итого $\mathcal{O}(q \cdot d \cdot (n + m))$.

Решение подзадачи 2

27 баллов

В этой подзадаче $d \leq 10^9$, но общее количество дорог и запросов все еще невелико — $n, m, q \leq 100$.

Будем рассматривать только интересные моменты, то есть те, когда Кирилл оказывается на перекрёстке. Если он стоит на перекрёстке (x_i, y_j) в момент t и t чётно, то следующий перекрёсток — (x_i, y_{j+1}) , до которого он дойдёт за $y_{j+1} - y_j$ секунд; если t нечётно — (x_{i+1}, y_j) за $x_{i+1} - x_i$ секунд. Если следующего перекрёстка нет или до него не успеть к моменту d , то оставшиеся $d - t$ секунд Кирилл идёт по прямой, и ответ находится сразу.

После каждого перехода один из индексов i, j увеличивается, поэтому переходов не больше $n + m$, и весь процесс моделируется за время $\mathcal{O}(q \cdot (n + m))$.

Решение подзадачи 3–4

70 баллов

Давайте моделировать только те перекрёстки, на которых происходит поворот (то есть до перекрёстка шли вправо, а потом вверх, или наоборот).

Заметим, что пройденный путь до некоторого перекрёстка (x_i, y_j) равен $(x_i - x_a) + (y_j - y_b)$, где (x_a, y_b) — стартовый перекрёсток. Пусть, например, из перекрёстка (x_i, y_j) мы пошли вверх, затем прошли перекрёстки $(x_i, y_{j+1}), \dots, (x_i, y_{j'-1})$, не поворачивая, и повернули вправо на перекрёстке $(x_i, y_{j'})$. Это означает, что:

- $(x_i - x_a) + (y_j - y_b)$ чётно, так как из (x_i, y_j) мы пошли вверх;
- $(x_i - x_a) + (y_{j''} - y_b)$ чётно для всех $j < j'' < j'$, так как на этих перекрёстках мы продолжили идти вверх;
- $(x_i - x_a) + (y_{j'} - y_b)$ нечётно, так как из $(x_i, y_{j'})$ мы пошли вправо.

То есть j' — это наименьший индекс, больший j , такой что чётность $y_{j'}$ отличается от чётности y_j . Аналогично, если из перекрёстка (x_i, y_j) мы пошли вправо, то повернём вверх на первой вертикальной дороге $x_{i'}$ ($i' > i$), чётность которой отличается от чётности x_i .

Таким образом, достаточно предсчитать массивы $j_x[i]$ ($1 \leq i \leq n$) и $j_y[j]$ ($1 \leq j \leq m$), где $j_x[i]$ — наименьший индекс $i' > i$ такой, что $x_{i'} \not\equiv x_i \pmod{2}$ (аналогично $j_y[j]$ для массива y). Если такого индекса нет, положим $j_x[i] = n + 1$ (соответственно $j_y[j] = m + 1$): в этом случае следующего поворота не будет, и Кирилл уходит по текущей дороге в бесконечность. Оба массива считаются одним проходом справа налево.

Дальше моделирование устроено как в подзадаче 2, только из перекрёстка мы переходим сразу к следующему повороту.

Решение подзадачи 5

100 баллов

Пусть (x_a, y_b) — стартовый перекрёсток. Построим две последовательности индексов: $a_0 = a$, $a_k = j_x[a_{k-1}]$ и $b_0 = b$, $b_k = j_y[b_{k-1}]$ (последовательность обрывается, как только индекс становится равным $n + 1$, соответственно $m + 1$).

Так как в момент $t = 0$ Кирилл идёт вверх, повороты происходят на перекрёстках (здесь пара (i, j) обозначает перекрёсток (x_i, y_j)) (a_0, b_0) , (a_0, b_1) , (a_1, b_1) , (a_1, b_2) , $\dots$, (a_k, b_k) , (a_k, b_{k+1}) , $\dots$, пока одна из последовательностей не оборвётся.

Иными словами, если мы найдём наибольшее k такое, что $x_{a_k} + y_{b_k} \leq x_{a_0} + y_{b_0} + d$, то после этого останется промоделировать $\mathcal{O}(1)$ шагов.

Такое k можно найти бинарным поиском за $\mathcal{O}(\log(n + m))$, если научиться быстро вычислять значения a_k и b_k для произвольного k . Покажем, как это сделать на примере a_k . Рассмотрим чётности элементов массива x ; пусть, например, получилась последовательность как на рис. 2.

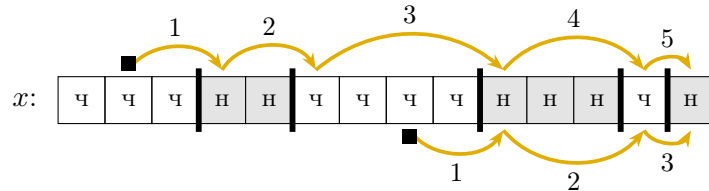


Рис. 2: Блоки чётности и прыжки. «ч» — чётная координата дороги, «н» — нечётная, жирные черты — границы блоков, чёрные квадраты — стартовые дороги двух цепочек прыжков. Цепочки начинаются в разных местах внутри блоков, но после того как они попадают в начало одного и того же блока, прыжки у них совпадают.

Видно, что значения делятся на блоки подряд идущих элементов с одинаковой чётностью и что все прыжки ведут в начало следующего блока. В частности, с какого бы места внутри блока мы ни стартовали, дальше будут посещены одни и те же начала блоков. Поэтому заведём массив $p_1 < \dots < p_{n'}$ индексов, с которых начинаются блоки (здесь n' — число блоков). После первого

прыжка $a_1 = j_x[a]$ мы попадём в начало некоторого блока p_s , после чего будут посещены дороги с индексами $p_{s+1}, p_{s+2}, \dots$, то есть $a_k = p_{s+k-1}$ при $k \geq 1$ (если $s + k - 1 > n'$, то последовательность уже оборвалась). Номер s находится за $\mathcal{O}(1)$, если для каждой дороги предпосчитать номер её блока, поэтому a_k (и аналогично b_k) вычисляется за $\mathcal{O}(1)$.

Итого получено решение со временем работы $\mathcal{O}(n + m + q \log(n + m))$.

Задача В. Цепочка из делителей

Все числа в массиве различны, поэтому будем ссылаться на элементы по их значениям. Для значения x обозначим $f(x)$ как длину самой длинной цепочки делителей, которая начинается в x . Тогда $s = \max f(x)$, а t — количество x , для которых $f(x) = s$.

Также будем поддерживать $\text{cnt}[k]$ ($1 \leq k \leq 20$) как количество x , для которых $f(x) = k$. Далее во **всех** решениях будет явно поддерживаться массив $f(x)$. Поэтому можно будет пересчитать $\text{cnt}[k]$ за $\mathcal{O}(1)$ для каждого изменения $f(x)$. По массиву cnt за $\mathcal{O}(1)$ легко восстановить s и t .

Решение подзадач 1–4

26 баллов

В этих подзадачах достаточно было научиться считать $f(x)$ наивно после каждого применённого запроса.

Для каждого значения x будем хранить его позицию $\text{pos}(x)$, причём, если происходит вставка в начало, то мы не пересчитываем все позиции, а новому вставленному элементу присваиваем позицию, которая на 1 меньше, чем позиция предыдущего первого элемента.

Ясно, что $f(x)$ можно вычислить по формуле, перебирая x в порядке убывания (максимум пустого множества равен 0):

$$f(x) = 1 + \max \left\{ f(k \cdot x) \mid 1 \leq k \leq \left\lfloor \frac{m}{x} \right\rfloor \text{ и } \text{pos}(k \cdot x) > \text{pos}(x) \right\}$$

Один такой пересчёт стоит $\sum \frac{m}{x} = \mathcal{O}(m \log n)$ по всем x из массива, так как $1 + \frac{1}{2} + \dots + \frac{1}{n} = \mathcal{O}(\log n)$. Этого достаточно для подзадач 1, 3 и 4. Для подзадачи 2 удобнее идти в обратную сторону: перебираем элементы справа налево, полагаем $f(x) = 1 + \text{best}[x]$ и для каждого делителя d числа x обновляем $\text{best}[d] \leftarrow \max(\text{best}[d], f(x))$. Тогда пересчёт стоит $\sum d(x) \leq 240 \cdot (n + q)$, где $d(x)$ — число делителей x .

Решение подзадач 5,7,10

20–40 баллов

Заметим, что при вставке либо удалении элемента x в начале массива изменяется только $f(x)$, и только его имеет смысл пересчитывать.

При удалении элемента x достаточно обнулить $f(x)$. А при его вставке придется потратить $\mathcal{O}(\frac{m}{x})$ времени на вычисление $f(x)$. Здесь начинает играть роль ограничение на то, что каждый элемент вставляется не более чем $c = 10$ раз.

Таким образом в худшем случае суммарная стоимость действий составит

$$c \cdot \frac{m}{1} + c \cdot \frac{m}{2} + \dots + c \cdot \frac{m}{q/c} = \mathcal{O}(c \cdot m \cdot \log \frac{q}{c})$$

Решение подзадачи 6

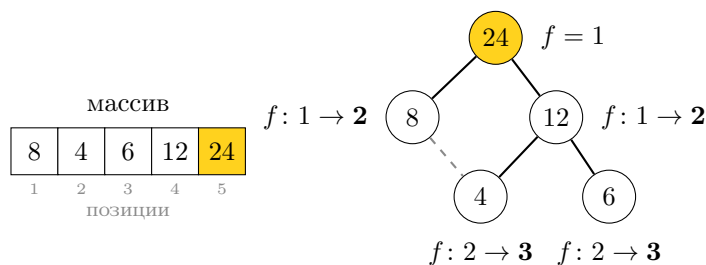
15–49 баллов

Положим, что x вставляется в конец. Тогда значение $f(y)$ может измениться только для тех y , которые являются делителями x .

Можно просто перебрать все y делители x и для каждого попытаться обновить $f(y)$ значением $f(x) + 1$, в случае если $\text{pos}(y) < \text{pos}(x)$. Причём при удачном обновлении y , возможно, придётся также рекурсивно обновить $f(z)$ для всех z делителей y и так далее.

Пример из условия: операция 1 24

Перед последней операцией массив равен $[8, 4, 6, 12]$, выполняем 1 24. Делители 24 в массиве: 12, 8, 6, 4 (и сам 24).



Ребро между числами означает, что меньшее делит большее (рёбра, следующие по транзитивности, не нарисованы). Пунктир 8–4: 4 делит 8, но 8 стоит *левее* 4, поэтому в $f(4)$ не учитывается.

При добавлении 24 обновляется значение $f(24) \leftarrow 1$, а через него обновляются значения $f(8) \leftarrow 2$ и $f(12) \leftarrow 2$.

Так как значение $f(8)$ обновилось, то через него теперь можно попробовать обновить значение $f(4)$, но этого делать нельзя, так как 4 стоит *правее* 8.

Но обновилось и значение $f(12)$, поэтому через него обновляются значения $f(6) \leftarrow 3$ и $f(4) \leftarrow 3$.

Решение подзадачи 8

17–60 баллов

Если есть только удаления с конца, то после k операций массив — это префикс длины $n - k$ исходного массива. Просто строим исходный массив вставками в конец (подзадача 6) и запоминаем ответ после вставки $(n - q)$ -го, $\dots$, n -го элемента. Ответы выводим в обратном порядке.

Решение подзадачи 9

25–65 баллов

Просто комбинация решения подзадач 5 и 6. Эти переходы никак друг другу не противоречат.

Решение подзадачи 11

36–75 баллов

Операции только на правом конце, то есть массив — это стек. Удаление последнего элемента в точности отменяет последнюю ещё не отменённую вставку, а между ними ничего не происходило. Поэтому при каждой вставке запишем в журнал все пары «(значение, старое f)», которые она изменила, а при удалении откатим их в обратном порядке. Исходный массив тоже строим вставками в конец, поэтому удаление его элементов обрабатывается так же. Суммарная работа по откатам не больше работы по вставкам.

Решение подзадачи 12

26–80 баллов

Только удаления с обоих концов. Развернём время: прочитаем все операции, найдём, какой массив останется в конце, и для каждой операции запомним, какое значение и с какой стороны она удалила. Пойдём с конца: возвращаем удалённые элементы на их места. Получаются только вставки с обоих концов, то есть подзадача 9. Как и в подзадаче 8, ответы при этом получаются в обратном порядке, и выводить их нужно с конца.

На данный момент мы умеем вставлять и извлекать элемент x из начала и делать вставку в конец массива. Осталось научиться только честно пересчитывать $f(x)$ при извлечении x из конца массива. Для этого давайте хранить величину:

- $g(x, i)$ ($1 \leq x \leq m$, $1 \leq i \leq 19$) — количество y , которые кратны x , правее x и $f(y) = i$.

Теперь при любом изменении $f(x)$ требуется пробежаться по его делителям y , которые находятся левее и обновить $g(y, \cdot)$.

Теперь при удалении элемента x справа заметим, что $f(y)$ может измениться только у его делителей, причём только на 1. Таким образом при уменьшении $f(x)$ на 1 (здесь $f(x)$ — старое значение) требуется для каждого делителя y левее x изменить $g(y, f(x))$ на -1 и $g(y, f(x) - 1)$ на $+1$, после чего если для какого-то делителя y оказалось, что $f(y) = f(x) + 1$ и $g(y, f(x)) = 0$, то надо уменьшить $f(y)$ на 1. Удаление самого x обрабатывается так же, как уменьшение $f(x)$ с 1 до 0, только $g(y, 0)$ не хранится и не меняется.

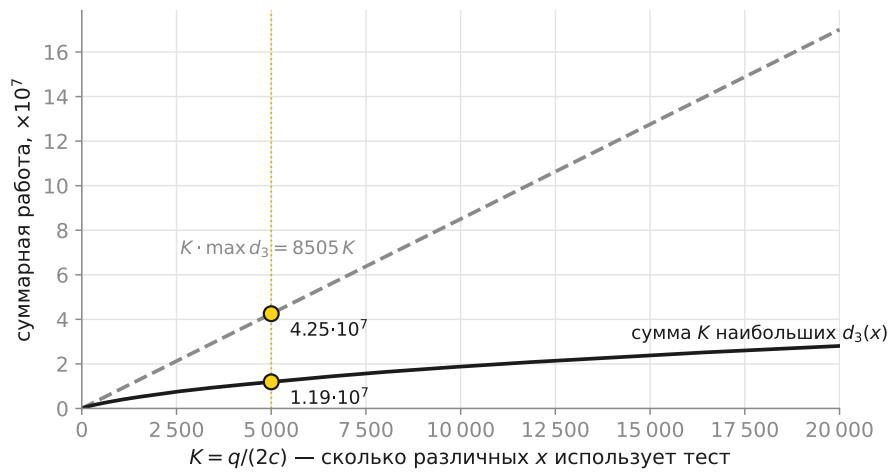
Этот процесс также требуется запускать рекурсивно для всех делителей x , в случае их изменения.

Наконец, оценим, сколько работает такая операция. В худшем случае при добавлении/удалении x в конец массива будут уменьшены все значения $f(y)$ по его делителям. И для каждого такого случая придется изменить значения $g(z)$ для всех делителей z . Таким образом в худшем случае эта операция работает за $d_3(x) = \#\{(a, b, e) \mid abe = x\}$ — количество способов разложить x в произведение трёх упорядоченных множителей (*функция делителей Пильца*, $d_3(x) = \sum_{y|x} d(y)$, где $d(y)$ — число делителей y).

Каждое значение вставляется не более c раз, а значит, и удаляется с конца не более c раз. Поэтому дорогих операций с одним и тем же x не больше $2c$, в худшем тесте участвует не более $K = \frac{q}{2c} = 5000$ различных x , и суммарное время работы не превосходит $2c$, умноженного на сумму K наибольших значений $d_3(x)$ среди $x \leq 10^6$. Грубая оценка $K \cdot \max d_3(x) = 8505 \cdot K$ здесь сильно завышена: чисел с большим $d_3(x)$ очень мало.

K	сумма K наибольших d_3	K -е по величине d_3	в среднем на одно x
10^2	$5.8 \cdot 10^5$	4860	5827
10^3	$3.8 \cdot 10^6$	2916	3794
$5 \cdot 10^3$	$1.19 \cdot 10^7$	1620	2386
10^4	$1.88 \cdot 10^7$	1134	1879
10^5	$6.1 \cdot 10^7$	252	613

При $K = 5000$ сумма наибольших значений равна $1.19 \cdot 10^7$, что в 3.6 раза меньше грубой оценки $4.25 \cdot 10^7$. Итого операции с концом массива в худшем случае стоят порядка $2c \cdot 1.19 \cdot 10^7 \approx 2.4 \cdot 10^8$ простых операций. Максимум $d_3(x)$ достигается при $x = 997\,920 = 2^5 \cdot 3^4 \cdot 5 \cdot 7 \cdot 11$ и равен 8505.



Итоговое время работы складывается из трёх частей:

- вставки в начало — $\mathcal{O}(c \cdot m \cdot \log \frac{q}{c}) \approx 10^8$, как в подзадаче 5;
- построение исходного массива вставками в конец — не больше суммы n наибольших d_3 , то есть $\leq 6.1 \cdot 10^7$;
- вставки и удаления в конце — $\approx 2.4 \cdot 10^8$ (см. выше).

Итого около $4 \cdot 10^8$ простых операций, что укладывается в ограничение в 5 секунд.

Задача С. Ремонт фонарей

Будем называть электриков и фонари *точками*, соответственно. Поскольку за секунду движется ровно один электрик, ответ — это **суммарное** расстояние, пройденное всеми электриками. Сразу заметим, что ответ не превосходит $2 \cdot 10^{18}$, а поэтому помещается в 64-битный целочисленный тип.

Сначала поймём, как выглядит оптимальное решение для набора точек на прямой.

- Каждый электрик, стартующий в x , обходит некоторый отрезок $[L, R] \ni x$. Это стоит не меньше $(R - L) + \min(x - L, R - x)$, и столько и достаточно: можно сначала сходить в ближний конец, вернуться, а потом сходить в дальний конец отрезка.
- Можно считать, что каждый промежуток между соседними точками проходит не больше одного электрика: если отрезки двух электриков пересекаются, один из них можно укоротить и это никогда не приведет к ухудшению ответа.

Решение подзадачи 1

5 баллов

Точек не больше 16. Пусть среди них e электриков и ℓ фонарей, $e + \ell = s \leq 16$. Можно перебрать, какому электрику достанется каждый фонарь: это e^ℓ вариантов, в худшем случае $6^{10} \approx 6 \cdot 10^7$. Для каждого варианта считаем стоимость каждого электрика по формуле выше. Но лучше сразу написать решение подзадачи 2.

Решение подзадачи 2

13 баллов

Раскроем все отрезки в отдельные точки (их $s \leq 2000$) и отсортируем. Электрики в оптимальном ответе берут фонари блоками подряд, причём в порядке своего расположения. Отсюда динамика $\text{dp}[i][j]$ — минимальная стоимость, если первые i электриков обслужили первые j фонарей. Переход: i -й электрик ничего не делает или берёт блок фонарей $(j', j]$ за стоимость по формуле. Это $\mathcal{O}(e \cdot \ell^2) = \mathcal{O}(s^3)$.

Есть мгновенная оптимизация до $\mathcal{O}(s^2)$: **электрику невыгодно заходить левее предыдущего электрика** (и, симметрично, правее следующего) — эти фонари дешевле отдать соседу.

Поэтому блок $(j', j]$ i -го электрика целиком лежит среди фонарей между $(i - 1)$ -м и $(i + 1)$ -м электриками. Пусть таких фонарей w_i . Тогда переходы для i -го электрика стоят $\mathcal{O}(w_i^2)$. Каждый фонарь попадает не больше чем в два таких окна, поэтому $\sum w_i \leq 2\ell$ и $\sum w_i^2 \leq (2\ell)^2$. Для j вне окна просто $\text{dp}[i][j] = \text{dp}[i - 1][j]$. Всего получается $\mathcal{O}(e \cdot \ell + \ell^2) = \mathcal{O}(s^2)$.

Решение подзадачи 3

22 балла

Точек до $2 \cdot 10^5$, нужна линейная динамика. Пройдём по точкам слева направо и для каждого промежутка между соседними точками будем отслеживать, *как* его проходит очередной электрик. Вариантов пять:

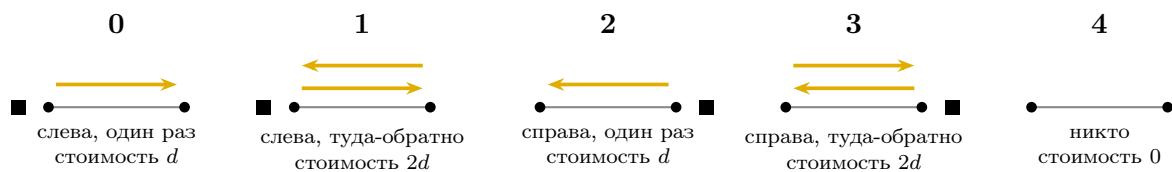


Рис. 3: Пять состояний промежутка длины d . Чёрный квадрат показывает, с какой стороны находится электрик, который проходит этот промежуток.

- 0 его проходит электрик слева, и его маршрут заканчивается правее (проходит один раз);
- 1 его проходит электрик слева, который сначала идёт вправо, потом возвращается и идёт влево (два раза);
- 2 его проходит электрик справа, маршрут которого заканчивается левее (один раз);
- 3 его проходит электрик справа, который сначала идёт влево, а потом возвращается (два раза);
- 4 промежуток никто не проходит.

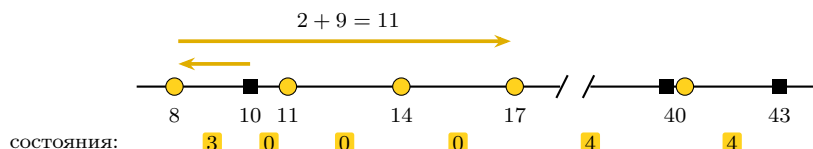
$dp[\text{состояние}]$ — минимальная стоимость уже пройденной части. Переходы через точку (состояние слева $\rightarrow$ состояние справа) такие. Во всех случаях, кроме фонаря в состоянии 4, можно остаться в том же состоянии.

точка	дополнительные переходы
фонарь	$0 \rightarrow 4, 1 \rightarrow 4$: здесь кончается маршрут, идущий слева; $4 \rightarrow 2, 4 \rightarrow 3$: здесь крайняя левая точка маршрута электрика справа; переход $4 \rightarrow 4$ запрещён : фонарь кто-то должен починить.
электрик	$4 \rightarrow 0$: идёт только вправо; $2 \rightarrow 4$: идёт только влево; $3 \rightarrow 0$: сначала влево, вернулся, затем вправо; $2 \rightarrow 1$: сначала вправо, вернулся, затем влево.

После точки прибавляем стоимость промежутка до следующей точки: d в состояниях 0, 2, $2d$ в состояниях 1, 3 и 0 в состоянии 4. Начинаем в состоянии 4, ответ — значение в состоянии 4 в конце. Точки, совпадающие по координате, обрабатываем как соседние с промежутком 0. Сложность $\mathcal{O}(s \log s)$ на сортировку и $\mathcal{O}(s)$ на динамику.

Второй пример из условия

$m = 3$, электрики в 10, 40, 43, фонари в 8, 11, 14, 17, 40.



Электрик из 10 сначала идёт влево (промежуток 8–10 в состоянии 3, стоимость $2 \cdot 2$), в точке 10 переходит $3 \rightarrow 0$, проходит промежутки до 17 по одному разу ($1 + 3 + 3$) и в фонаре 17 переходит $0 \rightarrow 4$. В точке 40 электрик переходит $4 \rightarrow 0$, промежуток длины 0 до фонаря 40 бесплатен, и фонарь переводит $0 \rightarrow 4$. Итого $4 + 7 = 11$.

Решение подзадачи 5

8–30 баллов

В этой подгруппе можно обойтись комбинаторными соображениями, но полное решение через матрицы (см. ниже) её тоже проходит: достаточно применить его при $m = 1$. Здесь лишь набросок того, как можно было набрать эти баллы без матриц, разобрав случаи руками.

При $m = 1$ отрезки — это подряд идущие целые точки, и раскрыть их нельзя: точек до 10^{18} . Но интересных ситуаций мало.

Упрощаем картинку. Фонари в точках, где есть электрики чинятся бесплатно — выкидываем их. Остальные фонари образуют *серии* подряд идущих точек. Из каждого отрезка электриков

нужны только крайние l_i и r_i : внутренним электрикам делать нечего, ведь заходить за соседнего электрика невыгодно (подзадача 2). Получаем отсортированный список электриков $E_1 < \dots < E_K$, $K \leq 2n$.

Каждый электрик одну сторону проходит дважды. Уйдя влево на a и вправо на b , он тратит $\min(2a + b, a + 2b)$. Пусть он заранее выбирает, какую сторону пройти дважды: у него появляются коэффициенты (слева, справа) = $(2, 1)$ или $(1, 2)$. Тогда электрики связаны только через промежутки между соседями, и по ним идёт динамика с двумя состояниями на электрика.

Промежуток между соседними электриками E_t и E_{t+1} . Если фонарей там нет, он бесплатен (сюда же попадают два электрика в одной точке). Иначе E_t идёт вправо, E_{t+1} — влево, и между ними остаётся один непройденный промежуток между соседними точками. Если x и y — насколько зашли электрики, а c_1, c_2 — их коэффициенты с этой стороны, это стоит $c_1x + c_2y$. Нужно выбрать лучший разрез; разбираем случаи:

- $c_1 = c_2$: стоимость равна $c_1(x + y)$, и выгодно пропустить *наибольший* промежуток;
- $c_1 \neq c_2$: наибольший промежуток уже может быть невыгоден (см. пример ниже). Но внутри серии подряд стоящих фонарей сдвиг разреза на одну точку даёт одному электрику $+1$, другому -1 , то есть меняет стоимость ровно на $c_1 - c_2$ — всегда в одну сторону. Значит, внутри серии выгоден один из её краёв, и кандидатов на разрез мало: промежутки между сериями и по два крайних шага в каждой серии. Их все и перебираем.

Фонари левее E_1 (правее E_K) обходит E_1 (E_K) со своим коэффициентом. Дальше — динамика $\text{dp}[t][\text{выбор } E_t]$ слева направо с переходом через промежуток (E_t, E_{t+1}) . Всё вместе — $\mathcal{O}((n + p) \log(n + p))$.

Пример, когда наибольший промежуток невыгоден

$m = 1$, электрики в точках 0 и 100, фонари в 50, 99 и 1000. Фонарь 1000 правее всех, его обходит электрик 100. Ему выгодно пройти правую сторону один раз, то есть выбрать коэффициенты $(2, 1)$: это стоит 900, а слева у него $c_2 = 2$. У электрика 0 слева ничего нет, он выбирает $(2, 1)$, и справа у него $c_1 = 1$. Варианты непройденного промежутка между 0 и 100:

непройденный промежуток	0–50	50–99	99–100
$1 \cdot x + 2 \cdot y$	$0 + 100$	$50 + 2 = 52$	$99 + 0$

Ответ $52 + 900 = 952$: электрик 0 доходит до 50, электрик 100 сходит к 99, возвращается и идёт к 1000. Обратите внимание: пропущен *не* наибольший промежуток 0–50, иначе вышло бы $100 + 900 = 1000$.

Переход к матрицам

Для подзадачи 6 и полного решения удобнее другой язык: перепишем линейную динамику из подзадачи 3 через матрицы. Переход через одну точку вместе с промежутком после неё — линейное преобразование в *min-plus* алгебре:

$$\text{dp}'_j = \min_i (\text{dp}_i + M_{ij}), \quad (A \otimes B)_{ij} = \min_k (A_{ik} + B_{kj}).$$

Это операция над матрицами 5×5 , она ассоциативна. Поэтому последовательность точек задаётся произведением матриц, а одинаковый повторяющийся кусок можно возводить в степень бинарным возведением.

Обозначим через $M(\text{тип}, d)$ матрицу точки данного типа с промежутком d после неё. Она

строится прямо по таблице переходов:

$$M_{ij} = \begin{cases} \text{cost}_j \cdot d, & \text{если переход } i \rightarrow j \text{ разрешён в точке этого типа,} \\ +\infty, & \text{иначе,} \end{cases}$$

где $\text{cost}_j \cdot d$ — стоимость промежутка длины d в состоянии j (d для $j = 0, 2$; $2d$ для $j = 1, 3$; 0 для $j = 4$). Типов три: фонарь, электрик и *пусто*. «Пусто» — это точка, где ничего нет: в ней разрешены только переходы $i \rightarrow i$ для всех пяти состояний (включая $4 \rightarrow 4$), то есть такая матрица просто добавляет стоимость промежутка.

Выпишем все три матрицы явно. Строка — состояние слева от точки (i), столбец — состояние справа (j), оба нумеруются 0, 1, 2, 3, 4 (номера подписаны серым); точкой « $\cdot$ » обозначено $+\infty$, конечные элементы выделены цветом. Во всех трёх матрицах столбец j содержит только $+\infty$ или $\text{cost}_j \cdot d$, то есть $d, 2d, d, 2d, 0$ по столбцам.

$M(\text{фонарь}, d)$					
	0	1	2	3	4
0	d	$\cdot$	$\cdot$	$\cdot$	0
1	$\cdot$	$2d$	$\cdot$	$\cdot$	0
2	$\cdot$	$\cdot$	d	$\cdot$	$\cdot$
3	$\cdot$	$\cdot$	$\cdot$	$2d$	$\cdot$
4	$\cdot$	$\cdot$	d	$2d$	$\cdot$

$M(\text{электрик}, d)$					
	0	1	2	3	4
0	d	$\cdot$	$\cdot$	$\cdot$	$\cdot$
1	$\cdot$	$2d$	$\cdot$	$\cdot$	$\cdot$
2	$\cdot$	$2d$	d	$\cdot$	0
3	d	$\cdot$	$\cdot$	$2d$	$\cdot$
4	d	$\cdot$	$\cdot$	$\cdot$	0

$M(\text{пусто}, d)$					
	0	1	2	3	4
0	d	$\cdot$	$\cdot$	$\cdot$	$\cdot$
1	$\cdot$	$2d$	$\cdot$	$\cdot$	$\cdot$
2	$\cdot$	$\cdot$	d	$\cdot$	$\cdot$
3	$\cdot$	$\cdot$	$\cdot$	$2d$	$\cdot$
4	$\cdot$	$\cdot$	$\cdot$	$\cdot$	0

Решение подзадачи 6

12–42 балла

Теперь отрезки одного типа не пересекаются. Пройдём по прямой, останавливаясь во всех концах отрезков. Между соседними остановками активно не больше одного отрезка электриков и одного отрезка фонарей, поэтому точки там образуют периодическую картину с периодом m и не больше чем двумя точками на период. Матрица одного периода — произведение не больше двух матриц точек, полные периоды возводим в степень, а неполные куски по краям проходим точка за точкой. Это та же идея, что в полном решении, только «период» здесь состоит всего из двух точек.

Решение подзадач 4 и 7

47–67 баллов

Общий случай. Запишем координату точки как $x = t \cdot m + \rho$, где $t = \lfloor x/m \rfloor$ — номер *слоя*, $\rho = x \bmod m$ — остаток. У всех точек одного отрезка остаток одинаковый, а слои образуют отрезок $[\lfloor l/m \rfloor, \lfloor r/m \rfloor]$.

Ключевая картинка

Нарисуем таблицу: строки — слои, столбцы — все $N = n + p$ отрезков, отсортированных по остатку. Отрезок занимает в своём столбце строки с $\lfloor l/m \rfloor$ по $\lfloor r/m \rfloor$. Тогда все точки прямой в порядке возрастания — это клетки таблицы, **прочитанные построчно, как текст**. Расстояние между соседними клетками строки равно разности остатков, а от последней клетки строки до первой клетки следующей — $\rho_1 + m - \rho_N$.

Каждой клетке сопоставим матрицу M (тип, промежуток до следующей клетки), где тип — «пусто», если отрезок в этом слое неактивен. Содержимое столбца меняется только в $2N$ *событиях*: отрезок включается в слой $\lfloor l/m \rfloor$ и выключается в слое $\lfloor r/m \rfloor + 1$.

Отсортируем события по паре (слой, столбец) и будем идти по таблице, неся вектор dp из пяти чисел. Сначала все клетки пусты, а $\text{dp} = (\infty, \infty, \infty, \infty, 0)$: до первой точки никто ничего

	$\rho = 1$	$\rho = 3$	$\rho = 6$	$\rho = 8$	
	+2	+3	+2	+3 ←	
слой 0		3			
слой 1	11	13			
слой 2	21	23	26		
слой 3	31	33	36		
слой 4	41		46	48	
слой 5	51		56	58	
слой 6	61		66	68	
слой 7	71		76	78	
слой 8			86	88	
слой 9				98	

фонарь
 электрик
 событие

} слои 5..7 одинаковы:
 $dp \leftarrow dp \otimes Q^3$

Рис. 4: $t = 10$; фонари на $[11, 71]$ и $[26, 86]$, электрики на $[3, 33]$ и $[48, 98]$. В клетках записаны координаты точек. Прямая читается построчно; вне событий строки повторяются.

не проходит. Указатель стоит на первой клетке слоя первого события. Обработка очередного события в клетке (слой t , столбец c) продвигает указатель до этой клетки, *не читая* её:

1. если t — слой указателя, просто читаем клетки строки от указателя до столбца c (не включая c);
2. иначе дочитываем текущую строку от указателя до конца; все строки строго между слоем указателя и t одинаковы, поэтому умножаем dp на Q^k , где Q — произведение матриц всей строки, а k — число таких строк; затем читаем строку t от начала до столбца c (не включая c);
3. меняем тип клетки c (отрезок включился или выключился). Сама клетка будет прочитана уже с новым типом.

Под «читаем клетку» понимается $dp \leftarrow dp \otimes M(\text{тип клетки, промежуток до следующей клетки})$. После последнего события все клетки снова пусты, а пустые клетки не меняют состояние 4, поэтому ответ — dp_4 в этот момент. Если хранить таблицу как массив из N матриц и каждый раз пересчитывать Q и куски строки в лоб, получится $\mathcal{O}(N)$ умножений на событие и $\mathcal{O}(N^2 \cdot 5^3 + N \log C \cdot 5^3)$ в сумме. Этого хватает при $N \leq 2000$.

Решение подзадач 8–10

100 баллов

Осталось быстро отвечать на запросы «произведение матриц на отрезке столбцов» и «изменить одну матрицу» — это классическое **дерево отрезков** над столбцами, в узлах которого хранятся произведения. Матрица Q целой строки — это просто корень дерева. Каждое событие обрабатывается за $\mathcal{O}(\log N)$ умножений в дереве и $\mathcal{O}(\log C)$ возведений в квадрат для Q^k . Итого $\mathcal{O}(N(\log N + \log C) \cdot 5^3)$.

Промежуточный вариант — корневая декомпозиция вместо дерева, $\mathcal{O}(\sqrt{N})$ на операцию. Она проходит подзадачу 8, но целая строка стоит у неё $\Theta(\sqrt{N})$ умножений.

В последних подзадачах важна константа. Что помогает:

- при чтении строки умножать *вектор* на матрицы узлов (25 операций), а не матрицы друг на друга (125 операций); матрицы перемножаются только при обновлениях и в возведении в степень;
- не заводить дерево над «удвоенной» строкой (массивом из $2N$ клеток, где строка записана дважды, чтобы кусок «от клетки до той же клетки следующей строки» был одним отрезком): это удваивает работу при обновлениях. Достаточно одного дерева над N столбцами,

а такой кусок — это суффикс строки плюс префикс, то есть два запроса к дереву;

- считать в `long long` с насыщением: при $INF = 4 \cdot 10^{18}$ сумма $INF + INF$ ещё помещается в `long long`, а `__int128` заметно медленнее;
- развернуть цикл min-plus умножения вручную.

Задача D. Посчитай экстремальные перестановки

Нетрудно заметить, что максимальное значение функции $f(p)$ равно $\lfloor n/2 \rfloor$.

Действительно в случае $f(p) > \lfloor n/2 \rfloor$, число $\lfloor n/2 \rfloor$ не может быть соседним ни с каким другим числом, так как разность между ними должна быть не меньше $1 + \lfloor n/2 \rfloor$.

Решение подзадачи 1

8 баллов

Перебираем все $n! \leq 40\,320$ перестановок, оставляем те, у которых $f(p) = \lfloor n/2 \rfloor$, а затем каждую сверяем с шаблоном a .

Решение подзадачи 2

8–16 баллов

При $n = 2k$ разделим элементы на малые и большие. Малыми будем называть числа от 0 до $k-1$, а большими от k до $n-1$. Ясно, что малые и большие числа в экстремальной перестановке должны чередоваться.

Пусть на позиции 0 стоит малое число, тогда малые числа занимают четные позиции, а большие — нечетные. Докажем по индукции, что на позициях $2j$ и $2j+1$ стоят числа $k-1-j$ и $2k-1-j$. Пусть позиции $0, \dots, 2j-1$ уже заняты. Соседи числа $k-1-j$ должны быть не меньше $2k-1-j$, а среди свободных больших чисел такое одно. На четной позиции правее $2j$ у него было бы два свободных соседа, поэтому оно стоит на позиции $2j$, а справа от него — $2k-1-j$.

Таким образом, если на позиции 0 стоит малое число, то экстремальная перестановка имеет вид $k-1, 2k-1, k-2, 2k-2, \dots, 1, k+1, 0, k$, а если большое, то это ее разворот $k, 0, k+1, 1, \dots, 2k-2, k-2, 2k-1, k-1$.

Остается сверить обе перестановки с шаблоном a .

Решение подзадачи 3

4–20 баллов

При четном n ответ всегда равен 2, разберем случай $n = 2k+1$.

Построим таблицу для малых значений n и попробуем найти закономерность.

n	2	3	4	5	6	7	8	9	10	11	12
ответ	2	6	2	14	2	32	2	72	2	160	2

Оказывается, что в случае $n = 2k+1$ ответ равен $2^{k-1}(k+5)$, почему ответ именно такой будет понятно далее.

Общая терминология для нечетного n

Положим, что $n = 2k+1$. Элементы меньше k будем называть малыми, а больше k будем называть большими.

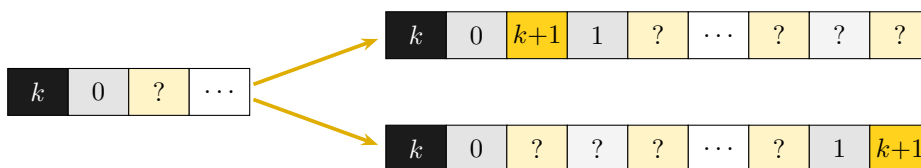
Ясно, что по обе стороны от k малые и большие значения обязаны чередоваться, а соседями значения k могут выступать только 0 и $2k$.

Ясно, что если в экстремальной перестановке первое значение равно k , то вторым может идти либо значение 0, либо значение $2k$ (не теряя общности будем считать, что это значение 0, второй случай обрабатывается заменой $a_i \mapsto 2k - a_i$). То есть перестановка имеет вид

k	0	?	?	?	...	?	?	?
-----	---	---	---	---	-----	---	---	---

При этом понятно, что малые и большие значения обязаны чередоваться, то есть после $2k$ стоит малое значение, затем большое, затем малое и так далее.

Рассмотрим значение $k + 1$ он может стоять только на третьей или последней позиции. Это происходит из-за того, что из непоставленных значений оно может соседствовать только с числом 1. Таким образом, оно будет стоять в паре с числом 1 либо в начале, либо в конце.



Далее рассмотрим пару значений $k + 2$ и 2, по тем же причинам $k + 2$ обязан стоять с краю отрезка, где значения еще не расставлены. И так далее. То есть пары значений $(1, k + 1)$, $(2, k + 2)$, $\dots$, $(k - 1, 2k - 1)$ ставятся парами. В конце этого процесса останется единственная позиция, на которую встанет число $2k$.

Таким образом, всего есть 2^{k-1} экстремальных перестановок, где $p_0 = k$ и $p_1 = 0$. Но требуется также соответствовать шаблону.

Соответствие шаблону можно обеспечить, если рассматривать динамику по подотрезку $\text{dp}[l, r]$ — количество способов совершить первые $\frac{n-2-r+l}{2}$ шагов алгоритма выше, чтобы получить полуинтервал непоставленных значений $[l, r)$ и так, чтобы вне этого полуинтервала все значения были поставлены верно. Такую динамику легко вычислить за время $\mathcal{O}(n^2)$.

Пример при $n = 9$

Покажем одну из $2^3 = 8$ возможных цепочек шагов. Знаком «?» отмечены непоставленные позиции (полуинтервал $[l, r)$), цвет показывает, малое или большое значение там должно стоять.

0	1	2	3	4	5	6	7	8	
4	0	?	?	?	?	?	?	?	$[2, 9)$
4	0	5	1	?	?	?	?	?	пара $(1, 5)$ слева, $[4, 9)$
4	0	5	1	?	?	?	2	6	пара $(2, 6)$ справа, $[4, 7)$
4	0	5	1	7	3	?	2	6	пара $(3, 7)$ слева, $[6, 7)$
4	0	5	1	7	3	8	2	6	последнее место — число $2k = 8$

Получилась перестановка $[4, 0, 5, 1, 7, 3, 8, 2, 6]$: модули разностей соседей равны 4, 5, 4, 6, 4, 5, 6, 4, все они не меньше $k = 4$. На каждом шаге динамика проверяет, что две только что поставленные позиции согласуются с шаблоном a .

Решение подзадачи 5

20–40 баллов

Теперь значение k может стоять на любой четной позиции в массиве. Если оно стоит на позиции 0, то это случай подзадачи 4, если на позиции $2k$, то к этому случаю можно свестись, развернув массив a .

В оставшемся случае k находится где-то посередине, не теряя общности будем считать, что слева от него стоит $2k$, а справа стоит 0 (обработка симметричного случая учитывается в замене a на свою развернутую версию). Снова можно заключить, что малые и большие значения обязаны чередоваться.

0	1	2	3	4	5	6	7	8	9	10	11	12
?	?	?	?	?	$2k$	k	0	?	?	?	?	?
$2L + 1$						$2R + 1$						

Пусть слева осталось $2L + 1$ не расставленных значений, а справа — $2R + 1$. Такие же рассуждения, как в подзадаче 4 показывают, что пары значений $(k + 1, 1), (k + 2, 2), \dots, (k + R, R)$ обязаны быть расставлены в правой половине массива, после чего останется единственная свободная позиция, на которую должно будет встать большое значение.

Рассмотрим левую половину, в которую будем расставлять пары $(2k - 1, k - 1), \dots, (2k - L, k - L)$ (точно также как и справа, только представляем симметрично относительно значений по k). В ходе этого процесса будет расставлено $L + R$ пар значений.

В конце процесса останется свободная позиция для малого числа слева и для большого числа справа, туда единственным способом встанут числа $R + 1$ и $k + R + 1$.

Пример при $n = 13$

Здесь $k = 6$, $L = R = 2$. Справа расставляются пары $(7, 1)$ и $(8, 2)$, слева — пары $(11, 5)$ и $(10, 4)$, каждую пару можно поставить к любому краю своего свободного отрезка. Покажем одну из $2^{L+R} = 16$ цепочек шагов.

0	1	2	3	4	5	6	7	8	9	10	11	12	
?	?	?	?	?	12	6	0	?	?	?	?	?	начало
?	?	?	?	?	12	6	0	7	1	?	?	?	справа: пара (7, 1)
?	?	?	?	?	12	6	0	7	1	?	2	8	справа: пара (8, 2)
?	?	?	11	5	12	6	0	7	1	?	2	8	слева: пара (11, 5)
4	10	?	11	5	12	6	0	7	1	?	2	8	слева: пара (10, 4)
4	10	3	11	5	12	6	0	7	1	9	2	8	числа 3 и 9

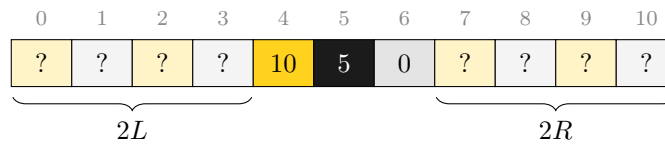
Заметим, что заранее известно, какие значения попадут в правую часть, а какие в левую. Поэтому можно посчитать ответ для каждой из половин независимо той же техникой, что и для подзадачи 4 и перемножить ответы.

В итоге получается решение со временем работы $\mathcal{O}(n^2)$.

Решение подзадачи 6

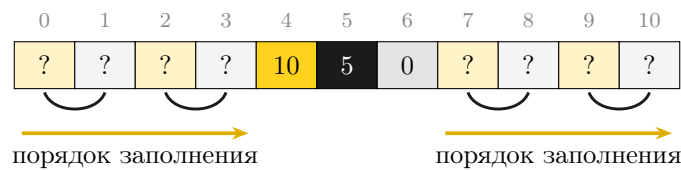
15–55 баллов

Разберем оставшийся случай, когда k находится на нечетной позиции.



В этом случае можно расставить пары значений $(k+1, 1), (k+2, 2), \dots, (2k-1, k-1)$ в таком порядке, причем очередная пара значений будет попадать либо в начало левой части, либо в начало правой части.

Таким образом все свободные позиции заранее однозначно делятся на пары так, что каждая поставленная пара значений будет поставлена на парную позицию. В каждой части пары позиций заполняются слева направо, выбор есть только в том, в какую часть пойдет очередная пара.



Таким образом, если части неизвестных значений содержат по $2L$ и $2R$ неизвестных позиций, то всего есть $\binom{L+R}{L}$ экстремальных перестановок, а количество подходящих под шаблон можно вычислить с помощью ДП $\text{dp}_2[l, r]$ — количество способов поставить первые $l+r$ пар так, что l из них попадут в левую часть, а r в правую, причем нигде на поставленных позициях не будет нарушен шаблон.

Пример при $n = 11$

Здесь $k = 5$, $L = R = 2$. Пары $(6, 1), (7, 2), (8, 3), (9, 4)$ ставятся по очереди, каждая — в первую свободную пару позиций левой или правой части. Всего $\binom{4}{2} = 6$ вариантов, покажем один из них.

0	1	2	3	4	5	6	7	8	9	10	
?	?	?	?	10	5	0	?	?	?	?	$(l, r) = (0, 0)$
?	?	?	?	10	5	0	6	1	?	?	пара $(6, 1)$ вправо, $(0, 1)$
7	2	?	?	10	5	0	6	1	?	?	пара $(7, 2)$ влево, $(1, 1)$
7	2	?	?	10	5	0	6	1	8	3	пара $(8, 3)$ вправо, $(1, 2)$
7	2	9	4	10	5	0	6	1	8	3	пара $(9, 4)$ влево, $(2, 2)$

Решение подзадачи 7

45–61 баллов

Объединяем решения предыдущих подзадач. Перебираем позицию центрального элемента и суммируем ответы, получилось решение за $\mathcal{O}(n^3)$. Ранее было много разговоров про симметричные случаи поэтому предъявим явно как их можно перебрать, не учитывая один и тот же ответ дважды

Перебор случаев при нечетном n

$\text{ans} \leftarrow 0$

повторить 2 раза:

для i от 1 до $n - 2$:

вид перестановки: $\dots \boxed{2k} \boxed{k} 0 \dots$, k на позиции i

если i четно: прибавить к ans ответ подзадачи 5

иначе: прибавить к ans ответ подзадачи 6

повторить 2 раза:

вид перестановки: $\boxed{k} 0 \dots$, прибавить к ans ответ подзадачи 4

заменить каждое известное значение шаблона: $a_j \leftarrow 2k - a_j$

развернуть массив a

ответ: ans

Внутренний повтор с заменой $a_j \leftarrow 2k - a_j$ разбирает оба вида начала: $[k, 0, \dots]$ и $[k, 2k, \dots]$ (после двух замен шаблон возвращается к исходному). Внешний повтор с разворотом массива добавляет случаи $[\dots, 0, k, 2k, \dots]$ в середине и k на последней позиции. Каждая экстремальная перестановка подходит ровно под один из разобранных случаев, поэтому ничего не посчитано дважды.

Решение подзадачи 8

58–74 баллов

В подзадаче 7 для каждого положения k запускалась динамика за $\mathcal{O}(n^2)$. Заменяем динамику комбинаторикой, тогда каждое положение k будет обрабатываться почти за линейное время.

Во всех трех случаях из подзадач 4–6 экстремальная перестановка задается последовательностью решений: пары $j = 1, 2, \dots$ ставятся по очереди, и каждый раз выбирается одна из двух сторон. Такую последовательность удобно изображать монотонным путем на клетчатой решетке. Точка (x, y) означает, что x пар уже ушло влево, а y — вправо. Шаг вправо по решетке ($x \rightarrow x + 1$) означает, что очередная пара пошла влево, шаг вверх ($y \rightarrow y + 1$) — что вправо. Пара j ставится шагом, который выходит из точки на диагонали $x + y = j - 1$.

Каждое известное значение шаблона говорит, *куда* и *когда* встала его пара, то есть требует, чтобы путь прошел через конкретное ребро решетки. Остается посчитать пути, проходящие через заданный набор ребер.

Пути через обязательные ребра

Число монотонных путей из точки (x_1, y_1) в точку (x_2, y_2) равно $\binom{(x_2 - x_1) + (y_2 - y_1)}{x_2 - x_1}$ (и 0, если $x_2 < x_1$ или $y_2 < y_1$). Если путь обязан пройти через ребра $e_1, e_2, \dots, e_t$ (отсортированные по номеру пары), то число таких путей равно произведению: от старта до начала e_1 , от конца e_1 до начала e_2 , $\dots$, от конца e_t до финиша. Если финиш не фиксирован, а годится любая точка на диагонали $x + y = c$, то последний множитель равен $2^{c - j_t}$, где j_t — номер последней обязательной пары.

Факториалы и обратные к ним по модулю $10^9 + 7$ предподсчитаем до n , тогда каждый биномиальный коэффициент считается за $\mathcal{O}(1)$. Ниже разберем, как известные значения превращаются в обязательные ребра в каждом из случаев.

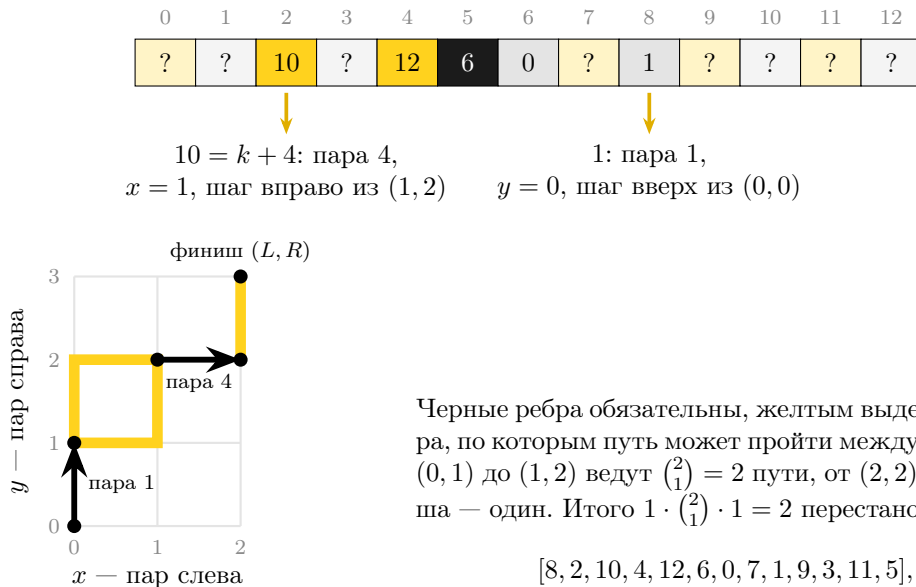
Случай 1: k на нечетной позиции (подзадача 6)

Это самый простой случай. Пусть k стоит на позиции i , слева от него $2k$, справа 0, в левой части $2L$ позиций, в правой $2R$. Путь идет из $(0, 0)$ в фиксированную точку (L, R) — это ровно состояния динамики $\text{dp}_2[l, r]$ из подзадачи 6. Пара, ушедшая влево x -й по счету (с нуля), занимает позиции $2x$ (большое значение) и $2x + 1$ (малое), а ушедшая вправо y -й по счету — позиции $i + 2 + 2y$ и $i + 3 + 2y$.

Поэтому известное значение v на позиции p однозначно задает ребро. Номер пары $j = v$ для малого и $j = v - k$ для большого. Если $p < i$, то $x = \lfloor p/2 \rfloor$, и путь обязан сделать шаг вправо из точки $(x, j - 1 - x)$. Если $p > i$, то $y = \lfloor (p - i - 2)/2 \rfloor$, и путь обязан сделать шаг вверх из точки $(j - 1 - y, y)$. Если значение стоит не на позиции своего типа (малое на большой позиции или наоборот) или на месте $k, 2k, 0$ стоит что-то другое, то вклад этого случая равен 0. Если известны оба значения одной пары, их ребра должны совпасть.

Пример: $n = 13, k = 6$ на позиции 5

Известны $a_2 = 10$ и $a_8 = 1$, остальные значения шаблона неизвестны. Рассматриваем расположение 12, 6, 0 на позициях 4, 5, 6. Здесь $L = 2, R = 3$, путь идет из $(0, 0)$ в $(2, 3)$.



Случай 2: k на позиции 0 (подзадача 4)

Пусть $p_0 = k, p_1 = 0$. Свободный отрезок — позиции $[2, n - 1]$, в него ставятся пары $j = 1, \dots, c$, где $c = k - 1$, а в последнюю свободную клетку встает число $2k$, назовем его *ником*. Пусть x пар уже ушло к левому краю отрезка и y — к правому. Пара j , поставленная слева, занимает позиции $2 + 2x$ (большое) и $3 + 2x$ (малое), поставленная справа — позиции $n - 1 - 2y$ (большое) и $n - 2 - 2y$ (малое). Путь начинается в $(0, 0)$ и заканчивается в любой точке диагонали $x + y = c$, а пик встает на позицию $2 + 2X$, где X — координата финиша.

Здесь появляется тонкость: по позиции известного значения не видно, с какой стороны оно поставлено. Например, большое значение на четной позиции p могло прийти слева ($x = (p - 2)/2$), а могло справа ($y = (n - 1 - p)/2$). Поэтому у каждой известной пары два кандидата на обязательное ребро (кандидат с отрицательной координатой сразу отбрасываем). Если известны оба значения пары, сторону определяет их порядок: слева большое стоит перед малым, справа — после.

Как выбрать сторону

Слева номера пар возрастают слева направо, справа — справа налево, а пик стоит между ними. Значит, если пары $j_1 < j_2$ стоят на позициях $p_1 < p_2$, то пара j_1 поставлена слева, а если $p_1 > p_2$ — справа.

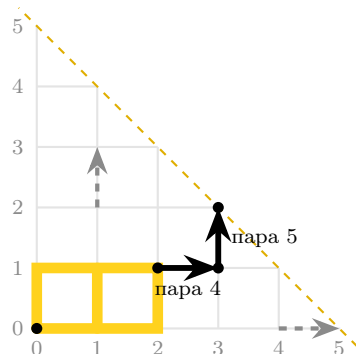
Сравнивая каждую известную пару со следующей известной (по номеру), узнаем сторону всех известных пар, кроме последней. Для последней перебираем оба варианта и складываем ответы: это разные пути, поэтому ничего не посчитано дважды. Если известна позиция пика q , то все еще проще: пары левее q поставлены слева, правее — справа, а финиш фиксирован: $X = (q - 2)/2$.

Пример: $n = 13$, $k = 6$ на позиции 0

Известны $a_7 = 4$ и $a_{10} = 11$. Пар $c = 5$, финиш — любая точка диагонали $x + y = 5$.

	0	1	2	3	4	5	6	7	8	9	10	11	12
	6	0	?	?	?	?	?	4	?	?	11	?	?
значение	позиция	пара	если слева		если справа								
4	7	4	$x = 2$: вправо из (2, 1)		$y = 2$: вверх из (1, 2)								
11	10	5	$x = 4$: вправо из (4, 0)		$y = 1$: вверх из (3, 1)								

финиш: $x + y = 5$



Пара 4 стоит левее пары 5 ($7 < 10$), значит, она поставлена слева: обязательный шаг вправо из (2, 1), путь приходит в (3, 1). Пара 5 — последняя, пробуем обе стороны. Шаг вправо из (4, 0) невозможен (из (3, 1) туда не попасть), шаг вверх из (3, 1) подходит. Черные ребра обязательны, желтым выделены ребра, по которым путь может пройти до них, серым пунктиром — отброшенные кандидаты. От (0, 0) до (2, 1) ведут $\binom{3}{2} = 3$ пути, после пары 5 свободных шагов не осталось ($2^0 = 1$). Ответ: 3.

Если дополнительно известно $a_8 = 12$ (пик), то $X = 3$, финиш — (3, 2), обе пары сразу определяются сравнением с позицией 8, и ответ тот же.

Случай 3: k на четной позиции в середине (подзадача 5)

Пусть k стоит на четной позиции i , слева от него $2k$, справа 0, в левой части $2L + 1$ позиций, в правой $2R + 1$. Как выяснено в подзадаче 5, правая часть заполняется ровно так же, как отрезок из случая 2: пары $j = 1, \dots, R$ и пик $k + R + 1$. Левая часть после разворота и замены $v \mapsto 2k - v$ устроена так же: соседнее с ней $2k$ превращается в 0, пары $(2k - j, k - j)$ — в пары $(k + j, j)$ для $j = 1, \dots, L$, а последнее малое $R + 1$ — в пик $k + L + 1$.

Поэтому каждую часть считаем как в случае 2 (каждую со своим путем, финишем на диагонали и правилом выбора стороны) и перемножаем ответы. Дополнительно проверяем, что каждое известное значение стоит в своей части: например, если $a_p = 1$ при $p < i$, то вклад равен 0.

Пример: $n = 13$, $k = 6$ на позиции 6

Известны $a_3 = 11$ и $a_{10} = 9$. Здесь $L = R = 2$.

0	1	2	3	4	5	6	7	8	9	10	11	12
?	?	?	11	?	12	6	0	?	?	9	?	?
левая часть						правая часть						

левая часть после разворота и замены $v \mapsto 12 - v$:

0	?	1	?	?	?
---	---	---	---	---	---

Правая часть. Известен только пик $9 = k + R + 1$ на позиции 10, то есть $X = 1$ и финиш — $(1, 1)$. Пути из $(0, 0)$ в $(1, 1)$: $\binom{2}{1} = 2$.

Левая часть. После преобразования $a_3 = 11$ превращается в малое значение 1 второй клетки отрезка, это пара 1. Слева она дает шаг вправо из $(0, 0)$, а справа ей нужно $y = 1$, то есть точка $(-1, 1)$, такого не бывает. Значит, обязательный шаг — вправо из $(0, 0)$, дальше остается один свободный шаг: $2^1 = 2$.

Ответ: $2 \cdot 2 = 4$. Левая часть — $[3, 10, 4, 11, 5]$ или $[4, 10, 3, 11, 5]$, правая — $[7, 1, 9, 2, 8]$ или $[8, 2, 9, 1, 7]$, и любой вариант слева сочетается с любым справа.

Время работы

Положений k всего $\mathcal{O}(n)$ (с учетом разворотов и замен $v \mapsto 2k - v$, как в подзадаче 7). Для каждого положения соберем известные значения нужной части, отсортируем их по номеру пары и пройдем по ним, перемножая биномиальные коэффициенты. Обработка одного положения занимает $\mathcal{O}(n \log n)$, всего $\mathcal{O}(n^2 \log n)$ на набор входных данных. При $n \leq 2000$ этого достаточно.

Решение подзадачи 9

100 баллов

Осталось избавиться от логарифма. Сортировка не нужна: если перебирать номера пар $j = 1, \dots, k - 1$ по порядку и смотреть в массив $\text{pos}[v]$ (позиция значения v в шаблоне или -1), то известные значения сразу получаются в порядке номеров пар. В случае 3 значения левой части перебираем в обратном порядке, так как замена $v \mapsto 2k - v$ разворачивает нумерацию пар.

Обработка одного положения k занимает $\mathcal{O}(n)$, всего $\mathcal{O}(n^2)$ на набор входных данных. При $n = 20\,000$ это порядка $4 \cdot 10^8$ простых операций, что укладывается в ограничение по времени.

Решение можно ускорить еще. Во-первых, для каждого положения k достаточно проходить только по известным значениям (один раз выписав их в порядке номеров пар), это $\mathcal{O}(m + 1)$, где m — количество известных значений. Во-вторых, положение k на позиции i в середине рассматривается, только если a_{i-1}, a_i, a_{i+1} совместимы с $2k, k, 0$, а таких i не больше $n - m + 3$. Итого $\mathcal{O}(n + (n - m) \cdot m)$, к тому же перебор можно прерывать, как только произведение стало нулевым.