# Всероссийская олимпиада школьников по информатике
# Региональный этап, II тур
# Разбор задач

Андрианов И. А.
Стрекаловский О. А.

Вологодский государственный университет
Факультет прикладной математики, компьютерных технологий и физики

Вологда
2016 г.

Чтобы минимизировать сумму квадратов, необходимо стараться выбирать числа $a, b$ и $c$ близкими к $n/3$.

Докажем сначала следующий факт:

- Пусть различные положительные целые $a < b$ таковы, что $a + b = m$.
- Тогда если сумма $a^2 + b^2$ минимальна, то $b - a \leq 2$.

Рассмотрим процесс «сближения» $a$ и $b$:
Пусть $b - a > 2$.
Тогда $a + 1 \neq b - 1$ и
$(a+1)^2 + (b-1)^2 = a^2 + 2a + 1 + b^2 - 2b + 1 =$
$a^2 + b^2 + 2(a-b) + 2 < a^2 + b^2 - 4 + 2 < a^2 + b^2 \Rightarrow$ взяв вместо $a$ и $b$ числа $a + 1$ и $b - 1$, мы получим два различных числа с такой же суммой, но меньшей суммой квадратов $\Rightarrow$ $a$ и $b$ нужно сдвигать до тех пор, пока $a + 1 \neq b - 1$ и этот процесс остановится, как только $b - a$ станет $\leq 2$.

- Рассмотрим теперь три числа $a$, $b$ и $c$, которые требуется найти в задаче.
- Применив предыдущее утверждение для $a$ и $b$ при фиксированном $c$, а также для $b$ и $c$ при фиксированном $a$, получим что все числа не более чем на 2 отстоят от значения $n/3$:

$$\begin{cases} b - a \leq 2 \\ c - b \leq 2 \end{cases} \Rightarrow c - a \leq 4 \Rightarrow a, b, c \in [n/3 - 2, n/3 + 2]$$

Переберем все тройки целых положительных чисел, где каждое число, не более чем на 2 отличается от $n/3$, и выберем среди них тройку с минимальной суммой квадратов.

Чтобы сумма квадратов была минимальна, нужно, чтобы числа отличались как можно меньше.

1. Если $(n-3) \mid 3$, то ответ — $\langle a, a+1, a+2 \rangle$, где $a = (n-3)/3$.
2. Если $(n-5) \mid 3$, то ответ — $\langle a, a+2, a+3 \rangle$, где $a = (n-5)/3$.
3. Иначе ответ — $\langle a, a+1, a+3 \rangle$, где $a = (n-4)/3$.

Один из вариантов точно подойдёт.

Вопросы по задаче?

# «Гипершашки»

1. Отсортируем массив `x` по возрастанию.
2. Создадим ассоциативный массив `count`, где `count[i]` — количество повторов числа `i`.
3. Создадим два упорядоченных массива `all` и `rep`, где:
   1. `all` — все входные числа без повторов;
   2. `rep` — входные числа, которые повторялись (но в массиве rep каждое будет в одном экземпляре).

# «Гипершашки»

Перебираем числа в ассоциативном массиве `count` — это будет наименьшее число в счёте.
Обозначим его за `a`.

- Если `count[a]` $\geq 3$, то добавляем к ответу 1.

# «Гипершашки»

Перебираем числа в ассоциативном массиве `count` — это будет наименьшее число в счёте.
Обозначим его за a.

- Если `count[a]` $\geq 2$, то рассмотрим вариант счёта с двумя экземплярами числа a и ещё одним числом в диапазоне $(a, a * k]$:
  1. Используем двоичный поиск по `all` для поиска количества чисел в диапазоне.
  2. Результат умножаем на 3, так как для каждого числа $x$ есть 3 варианта счёта: $\langle a, a, x\rangle$, $\langle a, x, a\rangle$, $\langle x, a, a\rangle$.

# «Гипершашки»

Перебираем числа в ассоциативном массиве `count` — это будет наименьшее число в счёте.
Обозначим его за a.

- Пробуем число a взять в одном экземпляре:
  - Вариант 1: оставшиеся два числа в счёте одинаковы.
    1. С помощью двоичного поиска по `rep` ищем количество чисел с повторами в диапазоне $(a, a * k]$.
    2. Результат умножаем на 3, так как 3 варианта: $\langle a, x, x\rangle$, $\langle x, x, a\rangle$, $\langle x, a, x\rangle$.
  - Вариант 2: оставшиеся два числа в счёте разные.
    1. С помощью двоичного поиска по `all` ищем количество.
    2. Используем формулу $C_2^n$, результат домножаем на 3 по аналогичной причине.

# «Гипершашки»

Существуют альтернативные решения для подсчёта суммы с использованием метода «двух указателей», что позволяет после сортировки посчитать ответ за $O(N)$, вместо $O(N \log N)$, как в варианте с двоичным поиском.

# «Гипершашки»

**Типичные ошибки**

- Разбор не всех возможных комбинаций.
- Выходы за границы массивов.

Вопросы по задаче?

Подзадача 1:

- Достаточно перебрать все числа из диапазона от $L$ до $R$ и для каждого из них проверить, является ли оно интересным.

Подзадача 2

- Достаточно перебирать только интересные числа, сохраняя в переборе уже поставленную часть числа и последнюю цифру.
- Будем дописывать только те цифры, которые приводят к сохранению свойства интересности.
- Между 1 и $10^{18}$ всего 4 686 824 интересных числа, поэтому перебор будет небольшой.

1. Пусть $c(L, R)$ — количество интересных чисел от $L$ до $R$.
2. Заметим, что $c(L, R) = c(1, R) - c(1, L - 1) \Rightarrow$ достаточно уметь вычислять только такие $c$, где первое число равно 1.
3. Для вычисления $c$ будем использовать метод динамического программирования.

Поскольку само число $10^k$ интересным не является, задача сводится к подсчету количества интересных чисел, состоящих из $k$ цифр, причем ведущие нули разрешаются (будет посчитано лишнее число 0, поэтому из ответа вычтем 1).

Пусть $d[i][j]$ — количество интересных чисел из $i$ цифр, последняя цифра которых равна $j$.

$$d[i][j] = \begin{cases} d[1][j] = 1 \text{ для всех } j \\ d[i][j] = sum(d[i-1][k], k = 0 \dots j) \text{ для } i > 1 \end{cases}$$

Вычисления необходимо производить по модулю $10^9 + 7$.

## Пример

1. Пусть i = 3, j = 5.
2. Нас интересует количество интересных чисел вида ab5.
3. Вместо ab можно подставлять двузначные интересные числа, которые кончаются на 0, 1, 2, 3, 4, 5 — суммируем их количество.

Для удобства можно сделать следующее (хотя можно решать и без этого).

1. Если $R$ – неинтересное число, то заменим его на наибольшее интересное число $< R$.
2. Для этого найдём в $R$ первую позицию $i$, где $R[i] > R[i+1]$.
3. Уменьшим $R[i]$ на 1, а все цифры после неё заменим на девятки.
4. Поскольку $R[i]$ уменьшилась на 1, то эта цифра может стать меньше предыдущей — тогда нужно будет сделать то же самое, и так несколько раз.

## Пример

Пусть $R = 24415$

1. Видим, что $1 > 4$ — меняем: $R` = 24399$.
2. Теперь стало $3 > 4$ — повторяем еще раз: $R`` = 23999$.
3. Мы получили наибольшее интересное число $\leq R$.

# «Интересные числа»
## Полное решение

- Если $R$ не является степенью 10, то не все интересные числа из такого количества цифр подходят.
- Если рассмотреть префикс интересного числа и если в нем есть хотя бы одна цифра, меньшая соответствующей цифры в $R$, то продолжение может быть любым.

### Пример

Пусть нас интересуют числа $T$ с префиксом 134.
$T$ = 13**4**xxxxx
$R$ = 13**6**77899
Так как $4 < 6$, то буквы x можно заменить на любые цифры $\geq 4$, идущие в неубывающем порядке.
Пусть нас интересуют числа $T$ с префиксом 135.
$T$ = 13**5**xxxxx
$R$ = 13**5**88899
Так как $5 = 5$, то так уже делать нельзя — можем выйти за $R$.

# «Интересные числа»

## Полное решение

1. Переберем длину общего префикса интересного числа и $R$. Пусть она равна $k$.
2. Для каждого значения $k$ запустим отдельное вычисление методом ДП:
   1. Пусть $d[i][j]$ — по–прежнему количество интересных чисел из $i$ цифр, в которых последняя цифра $j$.
   2. Добавляется условие: первые $k$ цифр префикса совпадают с соответствующими цифрами $R$, а следующая цифра — строго меньше.
   3. Формула для пересчета не меняется, а начальные значения меняются:
      $d[k+1][j] = 1$ для тех $j$, которые больше или равны $k$–й цифре числа $R$ и строго меньше его $(k+1)$-й цифры.

## Пример

Пусть $k = 3$.
$T$=135xxxxx
$R$=13588899
Первая цифра x может быть только 6 или 7, то есть
$d[3][6] = 1, d[3][7] = 1$.
Примечание: случай, когда этот x равен 8, будет автоматически при $k = 4$.

Вопросы по задаче?

# «Гармоничная последовательность»

- Пусть $x$ и $y$ — два первых члена последовательности.
  Тогда последовательность имеет вид:
  $x, y, (y - x), -x, -y, (x - y), x, y, \ldots$
  Длина периода равна 6.
- Нужно минимизировать функцию:
  $|x - b_1| + |y - b_2| + |(y - x) - b_3| + |-x - b_4| + |-y - b_5| + |(x - y) - b_6| + \ldots$
  Немного преобразуем её:
  $|x - b_1| + |y - b_2| + |(y - x) - b_3| + |x - (-b_4)| + |y - (-b_5)| + |(y - x) - (-b_6)| + \ldots$

# «Гармоничная последовательность»

$|x - b_1| + |y - b_2| + |(y - x) - b_3| + |x - (-b_4)| + |y - (-b_5)| +$
$|(y - x) - (-b_6)| + \ldots$

1. Теперь все слагаемые можно разбить на три группы:
   1. Вида $|x - a|$;
   2. Вида $|y - b|$;
   3. Вида $|(y - x) - c|$
2. Создадим три отдельных массива `bx`, `by` и `byx` для хранения значений вычитаемых элементов в этих группах, и отсортируем их по возрастанию.

# «Гармоничная последовательность»

Заметим, что функция является выпуклой вниз и по $x$, и по $y$, то есть имеет один минимум.

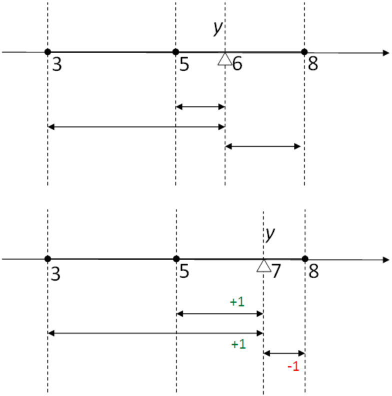
Это можно увидеть, например, из таких соображений:

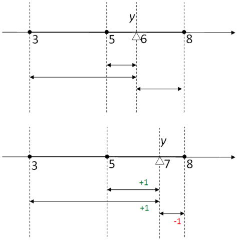
1. Зафиксируем конкретное $x$.
2. Тогда сумма по первой группе превратится в константу, а вторая и третья группы соединятся в одну со слагаемыми вида $|y - b|$.
3. Отсортируем их по возрастанию $b$:
   $|y - b_1| + |y - b_2| + |y - b_3| + \ldots$, где $b_1 \leq b_2 \leq b_3 \leq \ldots$.

# «Гармоничная последовательность»

Как изменится значение функции при увеличении $y$ на 1?

Пример: $|y-3|+|y-5|+|y-8|$






1. Обозначим $nLeft$ — количество точек слева от $y$, а $nRight$ — количество точек справа от $y$.
2. $f(x, y + 1) - f(x, y) = nLeft - nRight$.
3. При движении $y$ слева направо сначала $nLeft < nRight$, но с некоторого момента станет $nLeft > nRight \Rightarrow$ функция сначала убывает, потом возрастает.
4. Для фиксированного $y$ рассуждения будут аналогичны.

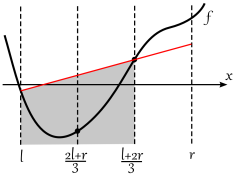
# «Гармоничная последовательность»

Соответственно можем использовать какой-нибудь численный метод оптимизации.
Например:

- Двумерный тернарный поиск;
- Метод градиентного спуска;

Такое решение может набрать 100 баллов.

Одномерный тернарный поиск:
Пусть функция $f(x)$ на отрезке $[l, r]$ имеет один минимум, и нужно его найти.




Алгоритм:

1. Посчитаем значения функции в точках $a = l + (r - l)/3, b = r - (r - l)/3$.
2. Если $f(a) < f(b)$, то абсцисса минимума $\in [l, b]$, иначе $\in [a, r]$.
3. Сужаем границы поиска и дальше повторяем аналогично.

Двумерный тернарный поиск — просто два вложенных одномерных тернарных поиска.
Значение $x$ ищет «внешний» тернарный поиск.
При каждом фиксированном $x$ задача превращается в одномерную, и значение $y$ ищет «внутренний» тернарный поиск.
Границы поиска: от $-2 \times 10^9$ до $2 \times 10^9$.

Можно воспользоваться префиксными и суффиксными суммами.

- Массив префиксных сумм для некоторого массива `v` устроен так:
  `pref[i] = v[0] + v[1] + ... + v[i]`
  Его несложно построить за линейное время.
- Аналогично, массив суффиксных сумм:
  `suf[i] = v[i] + v[i + 1] + ... + v[n]`.

- Чтобы найти сумму по первой группе, ищем двоичным поиском в bx такую позицию $pos$, что слева всем элементы $< x$, справа — $\geq x$.
- $Sum = x * pos - \texttt{pref[pos - 1]} + \texttt{suf[pos]} - x * (\texttt{bx.size()} - pos)$.
- Частные случаи: все элементы меньше $x$, все больше $x$.
- По остальным двум группам — аналогично.

# «Гармоничная последовательность»

Существуют и другие решения данной задачи, которые опираются на следующий факт:
Существует пара $(x, y)$, которая минимизирует сумму модулей, такая, что как минимум в двух из трех групп значения хотя бы одного модуля равны нулю.
Решение за $O(n^3)$:

1. Зафиксируем две из трех групп.
2. Переберем, какие именно модули равны нулю, получив тем самым $x$ и $y$, восстановим последовательность, обновим ответ.

Решение за $O(n^2 \log n)$:

1. Зафиксируем две из трех групп, переберем, какие именно модули равны нулю, получив тем самым $x$ и $y$.
2. Сосчитаем значение функции с помощью префиксных и суффиксных сумм и обновим ответ.

Вопросы по задаче?

Спасибо за внимание!