

# Челюсти

## Решение .

### Часть 1. Поиск следующей челюсти.

Конечно, если мы умеем находить номер челюсти и, наоборот, находить челюсть по её номеру, - а это второй и третий пункты задачи, - то решение первого пункта тривиально: просто находим челюсть со следующим номером. И в условиях олимпиады, вероятно, так и надо поступать. Но здесь я всё-таки порешаю это задание независимо от других. Просто чтобы продемонстрировать торжество метода. А метод уже применялся во второй задаче.

Я позволю себе автоцитату из решения второй задачи:

«...Чудесно. Мы уже кое-что знаем про число, которое нам нужно найти: оно до некоторого момента совпадает с заданным числом, а в первом же слева отличии его цифра должна быть больше соответствующей цифры заданного числа.

Второе. Искомое число должно быть при этом как можно меньше. Это означает, что первое слева несовпадение должно находиться как можно правее (поскольку первое несовпадение обязательно должно быть в сторону увеличения).»

Вот и будем двигаться по записи  $d_1d_2d_3\dots d_{n-1}d_n$  заданного числа  $X$  справа налево в поисках той цифры  $d_k$ , которую можно увеличить, не изменяя цифр слева от неё. Начинаем с  $k=n$ .

Если  $d_{k-1} > d_k$ , то мы можем увеличить  $d_k$  на 1 только в том случае, если  $d_{k-1} > d_k + 1$ , иначе число перестанет быть челюстью.

Если  $d_{k-1} < d_k$ , то мы можем увеличить  $d_k$  на 1 только в том случае, если  $d_k < 9$ , и число при этом останется челюстью.

Итого, если  $d_{k-1} > d_k + 1$  или  $d_{k-1} < d_k < 9$ , то увеличиваем  $d_k$ . В остальных случаях уменьшаем  $k$  (т.е. сдвигаемся влево) и повторяем проверку. Если процесс остановится на цифре  $d_k$ , то увеличиваем  $d_k$  на 1 и дописываем справа минимально возможный «хвост». Правило построения «хвоста» нам известно: каждый раз добавляем справа наименьшую возможную цифру, пока полученное число не станет больше  $X$ . В данном случае это правило задаёт такие действия:

если  $d_{k-1} < d_k$ , то хвост имеет вид 01010101...;

если  $d_{k-1} > d_k$ , то хвост начинается с цифры  $(d_{k-1} + 1)$ , а затем уже идёт 01010101... в необходимых количествах. Заметим, что в этом случае  $d_k < d_{k-1}$ , то есть  $(d_k + 1) \leq d_k \leq 9$ .

Что-то тут не то... Из описания процесса сразу видно, что если  $X$  –  $u$ -челюсть, то и следующее число будет  $u$ -челюстью, а если  $X$  –  $d$ -челюсть, то и следующее число будет  $d$ -челюстью. А ведь у нас есть челюсти и того, и другого вида. Когда же будет происходить смена вида челюсти? Ага, понятно: смена вида челюсти может произойти только тогда, когда у нас от числа  $X$  останется только две цифры, точнее, если нам не удалось остановиться на  $k=2$ . Значит, если мы дошли до того, что у нас осталось только две цифры, то надо писать какую-то хитрую обработку с несколькими вариантами. А не хочется..☺. И в данном случае нам повезло - можно обойтись без этого. Просто увеличиваем оставшееся двузначное число, пока оно не станет челюстью. Это много операций не займёт – двузначных челюстей довольно много, так что процесс не затянется. Да и наибольшую двузначную челюсть долго увеличивать не придётся – первая трёхзначная челюсть равна 101. Кстати, сразу виден и простой способ проверить, является ли полученное число челюстью – для этого достаточно, чтобы две последние цифры числа различались. И это правило выполняется и для всех двузначных чисел, и для числа 101, которое тоже может получиться, и – повезло – для однозначных чисел (просто первая невидимая цифра в них равна 0). В общем, не задача, а сплошное везение. Просто счастье какое-то...

## Часть 2. Считаем челюсти.

Эта задание тоже, в общем-то, сильно напоминает второе задание в задаче 2. Также надо подсчитать количество объектов (челюстей в данном случае), меньших данного. Только здесь считаем все меньшие челюсти, а не только челюсти такой же длины, как и заданная челюсть  $X$ .

Вот и хорошо. Пусть  $X = d_1 d_2 d_3 \dots d_{len-1} d_{len}$ . Подсчитаем сначала количество одно-, дву-, ... ,  $(len-1)$ -значных челюстей. А как это сделать? Довольно стандартно. Если сформулировать совсем коротко, то – увеличивая постепенно размер челюсти (выражаясь жаргонно, но точно: динамика по длине челюсти ☺). Если подробнее, то рассуждаем примерно так. Допустим мы уже знаем всё, что только нам придёт в голову о челюстях длины 1, 2, ...,  $(k-1)$ . А чтобы подсчитать  $k$ -значные челюсти заметим, что любая  $k$ -значная челюсть состоит из своей первой цифры, к которой приписана справа  $(k-1)$ -значная челюсть. Но ведь к данной первой цифре можно приписать не любую челюсть. Действительно, к цифре 4 нельзя приписать  $d$ -челюсть 3172, поскольку  $4 > 3 > 1$ , и нельзя приписать  $u$ -челюсть 7925, поскольку  $4 < 7 < 9$ . Понятно, что челюсть получится, если и только если к цифре 4 приписать любую  $d$ -челюсть, начинающуюся на 5, 6, 7, 8 или 9, либо любую  $u$ -челюсть, начинающуюся на 3, 2, 1 или 0. В общем случае, к цифре  $P$  можно приписать либо любую  $u$ -челюсть, начинающуюся на цифру, меньшую  $P$  (получая  $d$ -челюсть), либо любую  $d$ -челюсть, начинающуюся на цифру, большую  $P$  (получая  $u$ -челюсть). Вот и заведём себе величины  $ujaws(L, P)$  и  $djaws(L, P)$  – количество, соответственно,  $u$ -челюстей и  $d$ -челюстей длины  $L$ , начинающихся на цифру  $P$ . В связи с вышесказанным, сразу получаем формулы:

$$\begin{aligned}ujaws(L, P) &= djaws(L-1, P+1) + djaws(L-1, P+2) + \dots + djaws(L-1, 9), \\djaws(L, P) &= ujaws(L-1, P-1) + ujaws(L-1, P-2) + \dots + ujaws(L-1, 0), \\&L > 1, \quad 0 \leq P \leq 9.\end{aligned}$$

Также очевидно

$$ujaws(1, P) = djaws(1, P) = 1, \quad 0 \leq P \leq 9,$$

количество  $L$ -значных  $u$ -челюстей

$$ucount(L) = ujaws(L, 1) + ujaws(L, 2) + \dots + ujaws(L, 9),$$

а количество  $L$ -значных  $d$ -челюстей

$$dcount(L) = djaws(L, 1) + djaws(L, 2) + \dots + djaws(L, 9)$$

Обратите внимание, челюсти, начинающиеся на 0, в две последние суммы не входят – с 0 челюсти не начинаются. Тем не менее, величины  $ujaws(L, 0)$  и  $djaws(L, 0)$  нам нужны – ведь при подсчётах мы рассматриваем и челюсти, начинающиеся с 0. А с однозначными положительными числами всё ясно – все они являются челюстями.

Заполним массивы  $ujaws$ ,  $djaws$ ,  $ucount$  и  $dcount$ , определим  $len$  – длину заданной челюсти  $X$  и подсчитаем общее количество челюстей, длина которых меньше  $len$ . Затем добавим все челюсти длины  $len$ , первая цифра которых меньше первой цифры  $X$ .

И останется нам посчитать все челюсти длины  $len$ , начинающиеся на  $d_1$  и меньшие  $X$ .

Если  $X$  является  $u$ -челюстью, то сначала добавляем количество  $d$ -челюстей длины  $len$ , начинающиеся на  $d_1$  – понятно, что все они меньше  $X$ , ведь у всех у них вторая цифра меньше  $d_1$ , а у числа  $X$  вторая цифра больше  $d_1$ .

После этого остаётся подсчитать все  $u$ -челюсти длины  $len$ , начинающиеся на  $d_1$  и меньшие  $X$ .

Разделим  $u$ -челюсти длины  $len$ , начинающиеся на  $d_1$  и меньшие  $X$ , на два класса: те, у которых вторая цифра меньше  $d_2$ , и те, у которых вторая цифра равна  $d_2$ .

Каждой  $u$ -челюсти длины  $len$ , начинающейся на  $d_1$ , соответствует  $d$ -челюсть длины  $(n-1)$ , начинающаяся на цифру, большую  $d_1$  (такая челюсть получается отбрасыванием первой цифры). Следовательно, в первый класс входят

$$djaws(len-1, d_1+1) + djaws(len-1, d_1+2) + \dots + djaws(len-1, d_2-1)$$

челюстей.

Второй класс образуют все *u*-челюсти длины  $len$ , начинающиеся на  $d_1d_2$  и меньшие  $X$ . Каждой *u*-челюсти длины  $len$ , начинающейся на  $d_1d_2$ , соответствует *u*-челюсть длины  $(len-2)$ , начинающаяся на цифру, меньшую  $d_2$  (такая челюсть получается отбрасыванием первых двух цифр). Разделим все челюсти второго класса снова на 2 класса: те челюсти, у которых третья цифра меньше  $d_3$ , и те, у которых третья цифра равна  $d_3$ . И так далее... Пока не останутся челюсти длины 1. В общем, здесь слов приходится произносить гораздо больше, чем дел...

Если же  $X$  является *d*-челюстью, то выполняем в точности те же самые действия, что и в случае *u*-челюсти за исключением подсчёта *d*-челюстей длины  $len$ , начинающихся на  $d_1$ .

### Часть 3. Поиск челюсти с заданным номером $N$ .

Главное уже сделано – массивы *ujaws*, *djaws*, *ucount* и *dcount* уже заполнены. Осталась одна техника.

#### 3.1. Находим $len$ - длину челюсти.

Представьте себе, что все челюсти выписаны в ряд. Идём вдоль этого ряда и отсчитываем челюсти, пока не дойдём до номера  $N$ : сначала мы пройдем все челюсти длины 1, затем челюсти длины 2, длины 3 и т.д. На челюсть какой длины придётся номер  $N$  определить легко – ведь мы знаем, сколько имеется челюстей каждой длины.

#### 3.2. Находим первую цифру и тип челюсти.

Тоже несложно. Встали возле первой челюсти длины  $len$  и пошли вдоль ряда челюстей: проходим сначала все *d*-челюсти, начинающиеся на 1, затем *u*-челюсти, начинающиеся на 2, потом *d*-челюсти, начинающиеся на 1, *u*-челюсти, начинающиеся на 2, и т.д. На каком из перечисленных отрезков закончится наш отсчёт, тот и даёт нам первую цифру  $d_1$  и тип челюсти.

#### 3.3. Ищем вторую цифру.

Встанем на первую челюсть найденного типа, которая начинается на  $d_1$ . И пошли вдоль ряда челюстей. Если мы ищем *d*-челюсть, то сначала пойдут челюсти со второй цифрой 0, потом 1, потом 2 и т.д. Если же мы ищем *u*-челюсть, то сначала пойдут челюсти со второй цифрой  $(d_1+1)$ , потом  $(d_1+2)$  и т.д.

#### 3.4. Ищем следующую цифру.

А поиск каждой следующей цифры ничем не отличается от поиска второй цифры. Надо только не забыть, что если отбросить первую цифру *u*-челюсти, то останется *d*-челюсть, а если отбросить первую цифру *d*-челюсти, то останется *u*-челюсть.

### Часть 4. Подсчёт *u*-челюстей

Ну, а здесь двигаемся вдоль ряда челюстей точно также, как в части 3, пока не дойдём до  $N$ -ой челюсти. По дороге подсчитываем только *u*-челюсти, а *d*-челюсти пропускаем. В данном случае, видимо, проще посмотреть исходный текст программы – он довольно подробно откомментирован.