Разбор задачи “Bitty” XXV городской олимпиады по информатике города Петрозаводска

Итак, напомним коротко условие задачи, избавившись от легенды условия.

Дано натуральное n, необходимо посчитать количество чисел k от 1 до n таких, что: $k\ \ xor\ \ 2k\ \ xor\ \ 3k = 0$, где xor — операция исключающего или.

В данной задаче предполагалось решение с использованием динамического программирования по разрядам числа k в двоичном представлении. Давайте на время забудем о том, что у нас есть ограничение $k \leq n$. Понятно, что в таком случае ответ — бесконечность, но всё же получим для этого случая состояния, базу и переходы динамического программирования.

- Будем представлять число k в двоичной системе счисления.

- Вспомним, как умножать числа в столбик :). Чтобы умножить число k на c, надо умножать начиная с младших разрядов, двигаясь к старшим и храня “перенос” (сколько “лишнего” осталось при умножении в прошлом разряде). В нашем случае, надо не забыть, что умножение мы производим в системе счисления по основанию 2. Давайте опишем как умножать число 23 на 2 в двоичной системе счисления.

$$23_{(10)} = 10111(2),$$

$$\begin{array}{cr}
\scriptstyle\text{перенос} & \scriptstyle 1\ \ 1\,1\,1 \\
\hline
\times & 10111 \\
 & 2 \\
\hline
 & 101110
\end{array}$$

- Собственно, давайте генерировать число k начиная от младших к старшим разрядам. Поскольку нам нужно генерировать только такие k, что $k\ \ xor\ \ 2k\ \ xor\ \ 3k = 0$, то нам необходимо вычислять xor в каждом сгенерированном разряде для трех чисел: k, $2k$, $3k$, поэтому будем хранить переносы при умножении на 2 и на 3, обозначим их как $carry_2, carry_3$, соответственно.

- Таким образом, характеристиками состояния в динамике будут следующие $i, carry_2, carry_3$, где i — текущее количество поставленных разрядов. На самом деле мы здесь i никак использовать не будем, но оно понадобится для дальнейшего улучшения данной динамики с учетом ограничения $k \leq n$.

- Какие переходы в данной динамике? Пусть мы находимся в состоянии $(i, carry_2, carry_3)$, тогда понятно, что мы можем перебрать i-ый разряд d числа k от 0 до 1. Вычислить значение i-ого разряда в числах $2k, 3k$: $(2d + carry_2)\ mod\ 2, (3d + carry_3)\ mod\ 2$, соответственно. Проверить, что

  $$d\ \ xor\ \ ((2d + carry_2)\ mod\ 2)\ \ xor\ \ ((3d + carry_3)\ mod\ 2) = 0.$$

  Если это так, то можно перейти в новое состояние:

  $$(i + 1, \lfloor\frac{2d + carry_2}{2}\rfloor, \lfloor\frac{3d + carry_3}{2}\rfloor).$$

Данная динамика выглядит весьма стройно, за исключением того, что мы не определили базу динамики — сейчас данный алгоритм будет работать бесконечно, поскольку i всегда увеличивается. А базу динамики мы не определили, потому что сняли ограничение $k \leq n$. Чтобы решить исходную задачу, достаточно понять, как отслеживать условие, что сгенерированное число меньше либо равно n. Для этого необходимо:

- Перевести число n в двоичную систему счисления: $a_0, a_1, \ldots, a_{s-1}$, где s — длина числа n в двоичном представлении.

- Добавить в динамику еще один параметр $less$, который может принимать одно из трех значений

  - 0 — текущая сгенерированная запись числа k, содержащая i битов полностью совпадает с i младшими разрядами записи числа n,
  - 1 — текущая сгенерированная запись числа k, содержащая i битов строго меньше числа, представленного первыми i младшими разрядами числа n,
  - 2 — текущая сгенерированная запись числа k, содержащая i битов строго больше числа, представленного первыми i младшими разрядами числа n.

- Соответственно, при выборе нового разряда в переходе динамики новое значение параметра $less$ (new_less) зависит от того, меньше ли текущий разряд d числа k чем разряд a_i числа n:

  - если меньше, то $new\_less = 1$,
  - если больше, то $new\_less = 2$,

  - если равен, то $new\_less = less$.

- Теперь можно определить базу динамики: если $i \geq s, carry_2 = 0, carry_3 = 0$, то если $less = 2$ (то есть сгенерированное число больше n), вернуть 0, иначе — сгенерированное число k меньше либо равно n, вернуть 1.

Как и в любом алгоритме, необходимо составить его сложность. Единственное, в данной динамике, что осталось без обсуждения — это возможные значения параметров переноса $carry_2, carry_3$. Не трудно доказать, что $carry_2$ не может быть больше чем 1, а $carry_3$ не может быть больше чем 2. Для доказательства этого, просто необходимо рассмотреть наихудшее представление числа k — из одних единиц. Таким образом, асимтотика алгоритма составляет $O(\log(n))$.

Поскольку в задаче n было до 10^{18}, то для максимального теста, решение работало, грубо говоря, за $C \cdot 64, C = const$ операции. Так как задача была дана на школьную олимпиаду, ограничения были специально уменьшены, при такой сложности, можно было без затруднений дать $n \leq 10^{100000}$. Простое решение, которое перебирало k от 1 до n набирало 50 баллов.

Более подробно детали реализации можно посмотреть в решениях жюри.

Разбор подготовил Федулин Александр Андреевич, преподаватель Клуба творчества программистов ПетрГУ.