

# Недвоичная система счисления

## Решение

Имеется некоторое целое неотрицательное число  $N$ . Пусть  $a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0$  - его запись в недвоичной системе. Обозначим  $M$  число, записанное в недвоичной системе как  $a_n a_{n-1} a_{n-2} \dots a_2 a_1$ . Легко видно, что  $N = 2 \cdot M + a_0$  - наша недвоичная система не такая уж и не двоичная, ведь в ней сдвиг цифры на одну позицию вправо уменьшает «вклад» вдвое, как и в двоичной системе. Заметим, что  $(N - a_0)$  - чётное число, т.е. числа  $N$  и  $a_0$  имеют одинаковую чётность, причём  $a_0$  находится в диапазоне от 0 до 9. Если цифра  $a_0$  определена, то имеется столько же представлений числа  $N$ , сколько и числа  $M$ . Отсюда незамедлительно получаем рекуррентное соотношение

$F(N) = F((N-1)/2) + F((N-3)/2) + F((N-5)/2) + F((N-7)/2) + F((N-9)/2)$ , если  $N$  - нечётно,

$F(N) = F((N-0)/2) + F((N-2)/2) + F((N-4)/2) + F((N-6)/2) + F((N-8)/2)$ , если  $N$  - чётно,

где  $F(X)$  - количество различных способов записать число  $X$  в недвоичной системе.

Или, объединив два варианта, получим

$F(N) = F(\lfloor N/2 \rfloor) + F(\lfloor (N-2)/2 \rfloor) + F(\lfloor (N-4)/2 \rfloor) + F(\lfloor (N-6)/2 \rfloor) + F(\lfloor (N-8)/2 \rfloor)$ , или

$F(N) = F(\lfloor N/2 \rfloor) + F(\lfloor N/2 \rfloor - 1) + F(\lfloor N/2 \rfloor - 2) + F(\lfloor N/2 \rfloor - 3) + F(\lfloor N/2 \rfloor - 4)$ ,

где запись  $\lfloor x \rfloor$  означает, как обычно, целую часть числа  $x$  (или  $x$ , округленное вниз до целого числа).

Ура, ура, рекуррентное соотношение получено: заводим массив для хранения  $F$  и последовательно заполняем его слева направо - задача решена... Не тут-то было.  $2^{63}$  - это много. Это очень много. Даже если не обращать внимания на такие «мелочи» как память, всё равно, просто досчитать до  $2^{63}$  - никакого быстрогодействия не хватит. Что делать? Писать рекурсивную программу? Тоже не выход. Рекурсивный вызов здесь разделяется на пять вызовов, так что, похоже, рекурсия будет работать безумно медленно (так оно и оказывается при проверке). Написать динамически-рекурсивное решение? Есть такой стандартный приём - заполняем таблицу значений для небольших  $N$ , а для больших  $N$  вычисляем функцию рекурсивно, доводя рекурсию до уже вычисленных значений из таблицы. Дело немного ускоряется, позволяя в отведённые рамки времени/памяти обработать бОльшие значения  $N$ , но это увеличение не так существенно. Подумать? Интересная идея.

Довольно легко, просто глядя на формулу

$F(N) = F(\lfloor N/2 \rfloor) + F(\lfloor N/2 \rfloor - 1) + F(\lfloor N/2 \rfloor - 2) + F(\lfloor N/2 \rfloor - 3) + F(\lfloor N/2 \rfloor - 4)$ ,

увидеть, что  $F(2n+1) = F(2n)$ . Это наблюдение позволяет сократить потребляемую память вдвое, и время почти вдвое, или, соответственно, увеличить диапазон значений  $N$ , «влезających» в отведённый ресурс. Но и такое улучшение не решает дела.

Будем копать глубже... Именно глубже: распишем глубже нашу рекуррентную формулу. Обозначим  $N_1 = \lfloor N/2 \rfloor$ . Имеем

$F(N) = F(N_1) + F(N_1-1) + F(N_1-2) + F(N_1-3) + F(N_1-4)$ ,

А теперь обозначим  $N_2 = \lfloor N_1/2 \rfloor$  и выразим все слагаемые в последней формуле через  $F(N_2)$ ,  $F(N_2-1)$ ,  $F(N_2-2)$  и т.д. . Чтобы выписать получающуюся формулу потребуются рассмотреть два случая: чётное  $N_1$  и нечётное  $N_1$ , но это даже не нужно делать. Достаточно заметить, что в результате получится выражение  $F(N)$  в виде суммы семи чисел  $F(N_2)$ ,  $F(N_2-1)$ , ...,  $F(N_2-6)$ , взятых с некоторыми коэффициентами. Последнее слагаемое будет именно  $F(N_2-6)$ , поскольку  $\lfloor (N_1-4)/2 \rfloor - 4 = \lfloor N_1/2 \rfloor - 6 = N_2 - 6$ .

Повторим сделанный шаг. Обозначим  $N_3 = \lfloor N_2/2 \rfloor$  и выразим величины  $F(N_2)$ ,  $F(N_2-1)$ , ...,  $F(N_2-6)$  через  $F(N_3)$ ,  $F(N_3-1)$ ,  $F(N_3-2)$  и т.д. . В результате получится выражение  $F(N)$  в виде суммы восьми чисел  $F(N_3)$ ,  $F(N_3-1)$ , ...,  $F(N_3-7)$ , взятых с некоторыми коэффициентами. Последнее слагаемое будет именно  $F(N_3-7)$ , поскольку  $\lfloor (N_2-6)/2 \rfloor - 4 = \lfloor N_2/2 \rfloor - 7 = N_3 - 7$ .

Количество слагаемых увеличивается – это плохо. Но рост замедлился – этот вселяет надежды. Повторим шаг ещё раз: обозначим  $N_4 = \lfloor N_3/2 \rfloor$  и выразим величины  $F(N_3)$ ,  $F(N_3-1)$ , ...,  $F(N_3-7)$  через  $F(N_4)$ ,  $F(N_4-1)$ ,  $F(N_4-2)$  и т.д. . В результате, при чётном  $N_3$  получится выражение  $F(N)$  в виде суммы девяти чисел  $F(N_4)$ ,  $F(N_4-1)$ , ...,  $F(N_4-8)$ , взятых с некоторыми коэффициентами, поскольку при чётном  $N_3$   $\lfloor (N_3-7)/2 \rfloor - 4 = \lfloor N_3/2 \rfloor - 8 = N_4 - 8$ . Заметим, что при нечётном  $N_3$  слагаемых получится 8. При любых обстоятельствах, количество слагаемых не превосходит девяти. Можем сказать, что и при нечётном  $N_3$  слагаемых будет 9, и не беда, что одно из них входит в сумму с коэффициентом 0.

Ещё разок... Но это последний. Честное слово. Обозначим  $N_5 = \lfloor N_4/2 \rfloor$  и выразим величины  $F(N_4)$ ,  $F(N_4-1)$ , ...,  $F(N_4-8)$  через  $F(N_5)$ ,  $F(N_5-1)$ ,  $F(N_5-2)$  и т.д. . В результате получится выражение  $F(N)$  в виде суммы девяти (!) чисел  $F(N_5)$ ,  $F(N_5-1)$ , ...,  $F(N_5-8)$ , взятых с некоторыми коэффициентами, поскольку  $\lfloor (N_4-8)/2 \rfloor - 4 = \lfloor N_4/2 \rfloor - 8 = N_5 - 8$ . Ура! Снова девять слагаемых! Вон сколько восклицательных знаков я поставил. Но есть повод: количество слагаемых не увеличилось. Это означает, что оно не будет нарастать и дальше. И при каждом шаге у нас будет оставаться 9 слагаемых, а то и меньше, но это не страшно – как и выше, можем считать, что часть слагаемых входят в сумму с коэффициентом 0.

Итак, запишем "вычищенное", формальное решение. Вычисляем сумму

$$a_0 \cdot F(N) + a_1 \cdot F(N-1) + a_2 \cdot F(N-2) + \dots + a_8 \cdot F(N-8)$$

В начале полагаем  $a_0=1$ ,  $a_k = 0$ ,  $k=1, 2, \dots, 8$ .

Заменяем в этой сумме каждое слагаемое  $F(X)$  на  $F(\lfloor X/2 \rfloor) + F(\lfloor X/2 \rfloor - 1) + F(\lfloor X/2 \rfloor - 2) + F(\lfloor X/2 \rfloor - 3) + F(\lfloor X/2 \rfloor - 4)$  и получаем выражение

$$b_0 \cdot F(M) + b_1 \cdot F(M-1) + b_2 \cdot F(M-2) + \dots + b_8 \cdot F(M-8), \text{ где } M = \lfloor N/2 \rfloor$$

Полагаем  $N=M$ ,  $a_k = b_k$ ,  $k=0,1,\dots,8$  и повторяем процесс до тех пор, пока не дойдём до  $N=0$ .

Естественно положить здесь  $F(X)=0$  при  $X<0$ ,  $F(0)=1$  (0 записывается единственным способом), и тогда вся заключительная сумма равна  $a_0$ . Это и есть ответ на вопрос задачи.

Понятно, что каждый шаг требует константного времени, всего шагов получается  $O(\log N)$ , т.е. сложность алгоритма равна  $O(\log N)$ .