Рекурсия

Постановка задачи

Функция $f(n)$ задаётся рекуррентной формулой

$$\begin{aligned} &f(0) = 1, \\ &f(1) = 3, \\ &f(n) = 5\,f(n-1) + f(n-2) \quad \text{для } n \geq 2. \end{aligned}$$

В массив A сохраняются только нечётные значения $f(n)$. Требуется вычислить $A[39]$, то есть 40-й по счёту нечётный элемент последовательности $f(0), f(1), f(2), \ldots$.

Алгоритм

Посчитаем f(58).

Корректность

Заметим, что $f(0) = 1$ и $f(1) = 3$ — оба нечётные. Дальнейшая формула

$$f(n) = 5\,f(n-1) + f(n-2)$$

даёт следующие наблюдения по чётности:

- Если $f(n-1)$ и $f(n-2)$ оба нечётны, то сумма $5\,f(n-1)+f(n-2)$ есть чётное число.
- Если ровно одно из $f(n-1)$, $f(n-2)$ нечётно, а другое чётно, то $f(n)$ нечётно.
- Если оба предыдущих значения чётные, то $f(n)$ тоже чётно.

Отсюда следует, что чётность последовательности $f(n)$ повторяется с периодом 3: *нечётное, нечётное, чётное, нечётное, нечётное, чётное, ...* . В каждом таком трёхэлементном блоке первые два значения нечётны, третий — чётный.

Поскольку в одном блоке по два нечётных, чтобы получить ровно 40 нечётных, требуется 20 полных блоков. Первый блок охватывает $n = 0, 1, 2$, второй — $n = 3, 4, 5$ и т. д. Двадцатый блок соответствует индексам $n = 57, 58, 59$. В этом блоке $f(57)$ и $f(58)$ нечётные, а $f(59)$ чётное. Следовательно, 40-й нечётный элемент есть $f(58)$.

P.s.

Хотелось написать задачу на C++, но численное значение $f(58)$ слишком велико ($\geq 5^{56} > 2^{112} > 2^{64}$), чтобы поместиться в 64-битный тип. Поэтому в реализации используется Python.

P.p.s.

Если бы нужно было просто выбрать ответ из четырёх вариантов (где гарантированно присутствует правильный), можно обойтись без фактического расчёта. Достаточно заметить:

1. $f(58)$ нечётно.

2. При этом $f(n) \equiv f(n-2) \pmod 5$, поэтому $f(58) \equiv f(0) = 1 \pmod 5$.

То есть надо выбрать число оканчивающееся на 1. Ну, а такое ровно одно.