# АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
# ОБОБЩЕН ФИБОНАЧИ

Пресмятане на $n$-тия член на редицата може да се прави по различни начини. Най-неефективният от тях е да се използва директната рекурсия, заради многократното изчисляване на вече изчислени елементи ($O(1.6)^n$). По-добре е да се запомнят елементите в масив – линейно, но паметоемко. Без загуба на такива ресурси е алгоритъмът, при който всеки елемент се получава като сума от двата предишни, след което по-малкият от предишните номера вече не е нужен, неговото място се заема от по-големия предишен, а мястото на предишния се заема от току-що изчисления. Тези изчисления могат да се извършват и в $p$-ична бройна система, но това не е толкова ефективно, колкото ако входните данни се превърнат в десетична бройна система (реално машината ще работи в двоична), всичко се извършва със стандартните типове и накрая резултатът се върне обратно в $p$-ична бройна система.

Сравнително лесно може да се съобрази, че не е нужно да се пазят всички цифри на членовете: резултатът, който трябва да се изведе, позволява запомнянето само на последните две цифри от записите на $f_i$, следователно всички изчисления могат да се извършват по модул $p^2$.

Това са наивните идеи за решаване. В зависимост от реализацията, за тях са предвидени не повече от 30% от точките.

Известен е алгоритъмът със сложност $O(\log(n))$ за намирането на $n$-тия член на редицата на Фибоначи, основан на т. нар. „бързо повдигане на степен“. Наистина, ако се започне от матрицата $F_k = \begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix}$ и се извършва стандартно умножение на матрици с $E = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, получаваме $\begin{pmatrix} f_{k+1} & f_k \\ f_k & f_{k-1} \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} f_{k+1} + f_k & f_{k+1} \\ f_k + f_{k-1} & f_k \end{pmatrix} = \begin{pmatrix} f_{k+2} & f_{k+1} \\ f_{k+1} & f_k \end{pmatrix} = F_{k+1}$. След като пресметнем третия член на обобщената редица на Фибоначи $f_3 = f_1 + f_2$, можем да започнем изчислителния процес от $F_2 = \begin{pmatrix} f_3 & f_2 \\ f_2 & f_1 \end{pmatrix}$. Тогава ще имаме $F_{n-1} = \begin{pmatrix} f_n & f_{n-1} \\ f_{n-1} & f_{n-2} \end{pmatrix} = \begin{pmatrix} f_3 & f_2 \\ f_2 & f_1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-3} = F_2 E^{n-3}$. Търсеният член на редицата ще бъде в горния ляв ъгъл на матрицата, получена при умножението на $F_2$ с $n$-3-тата степен на матрицата $E$. Разбира се, всички изчисления се извършват по модул $p^2$. За реализации, свързани само с тази идея, са предвидени до 70% от точките. Допълнително определяне на цикличност на $E^s$ по модул $p^2$ може да доведе тази идея и до пълно решение.

Предлагаме идея, която може да реши задачата в максималните граници, без да ползва такива познания. Тя е свързана със съобразяването, че търсената цифра (по-точно – двойка цифри: предпоследната и последната) няма как да не е циклична. Наистина, за последните две цифри в редицата няма повече от $p^2$ възможности (всъщност, някои от тях може да не се реализират). Тогава в първия момент, в който и двата члена на редицата, пораждащи следващия, имат същите остатъци при деление на $p^2$ (в същия ред), които са имали на някой предишен етап, цикълът ще започне отново. Тъй като редицата е линейно разширяема надолу, със сигурност ще се повтори именно началният етап. По принципа на Дирихле, този цикъл няма как да е по-дълъг от $p^4$. Алгоритъмът се състои в определяне на дължината $d$ на такъв цикъл за зададените $p$, $f_1$ и $f_2$, намиране на остатъка $m$ от делението на входното $n$ на $d$, определяне на остатъка $r$ от делението съответния член на редицата с $p^2$ и извеждане на старшата цифра на $r$ в $p$-ична бройна система (нула, ако $r$ се записва само с една $p$-ична цифра).

*Автор: Павлин Пеев*