There is a clever algorithm for computing the Fibonacci numbers in a logarithmic number of steps. Recall the transformation of the state variables a and b in the `fib-iter` process $a \leftarrow a + b$ and $b \leftarrow a$. Call this transformation T, and observe that applying T over and over again n times, starting with 1 and 0, produces the pair $\mathrm{Fib}(n + 1)$ and $\mathrm{Fib}(n)$. In other words, the Fibonacci numbers are produced by applying T^n , the n^{th} power of the transformation T, starting with the pair $(1, 0)$. Now consider T to be the special case of $p = 0$ and $q = 1$ in a family of transformations T_{pq}, where T_{pq} transforms the pair (a,b) according to $a \leftarrow bq + aq + ap$ and $b \leftarrow bp + aq$. Show that if we apply such a transformation T_{pq} twice, the effect is the same as using a single transformation $\mathrm{T}_{p'q'}$ of the same form, and compute p' and q' in terms of p and q. This gives us an explicit way to square these transformations, and thus we can compute T^n using successive squaring, as in the `fast-expt` procedure. Put this all together to complete the following procedure, which runs in a logarithmic number of steps:

$\mathrm{Fib}(1) = 1$, a
$\mathrm{Fib}(0) = 0$, b

$a \leftarrow a + b$
$b \leftarrow a$
The above is T, so after T^n, $a = \mathrm{Fib}(n+2)$, $b = \mathrm{Fib}(n)$

Special case T_{pq} where T_{pq} transforms the pair (a,b)
$a \leftarrow bq + aq + ap$
$b \leftarrow bp + aq$

When p = 0 and q = 1
$a \leftarrow b(1) + a(1) + a(0)$
$b \leftarrow b(0) + a(1)$
$=$
$a \leftarrow a + b$
$b \leftarrow a$

Now we show that if we apply transformation T_{pq} twice, the effect is the same as using a single $\mathrm{T}_{p'q'}$ transformation, and we compute p' and q' in terms of p and q - thus giving us the ability to compute T^n using successive squaring (helping us achieve a logarithmic number of steps).

$T_{pq}(a, b)$
$T_{pq}((bq + aq + ap), (bp + aq))$ **applied once**
$= ((bp + aq)q + (bq + aq + ap)q + (bq + aq + ap)p, (bp + aq)p + (bq + aq + ap)q)$ **applied twice**
$= (bpq + aq^2 + bq^2 + aq^2 + apq + bpq + apq + ap^2, bp^2 + apq + bq^2 + aq^2 + apq)$
$= (b(q^2 + 2pg) + a(q^2 + 2pg) + a(p^2 + q^2), b(p^2 + q^2) + a(q^2 + pq))$
$T_{pq}(T_{pq}(a,b)) = T_{p'q'}(a,b) = (bq' + aq' + ap', bp' + aq')$
$p' = p^2 + q^2$
$q' = q^2 + 2pg$