Modular exponentiation with repeated squaring

```
Mod-Exp(a, b, n)
1   c = 0
2   d = 1
3   let <bk, bk-1, ..., b0> be the binary representation of b
4   for i = k downto 0
5       c = 2c
6       d = (d · d) mod n
7       if bi == 1
8           c = c + 1
9           d = (d · a) mod n
10  return d
```

If the inputs a, b, and n are β-bit numbers, then the total number of arithmetic operations required is $O(\beta)$ and the total number of bit operations required is $O(\beta^3)$. Example: $2^{(0)_2} \equiv 1 \pmod{10}$, $2^{(1)_2} \equiv 2 \pmod{10}$, $2^{(10)_2} \equiv 2^2 \equiv 4 \pmod{10}$, $2^{(100)_2} \equiv 4^2 \equiv 6 \pmod{10}$, $2^{(101)_2} \equiv 2 \cdot 6 \equiv 2 \pmod{10}$, $2^{(1010)_2} \equiv 2^2 \equiv 4 \pmod{10}$.