computing powers by repeated squaring

Canonical name	ComputingPowersByRepeatedSquaring
Date of creation	2013-03-22 19:18:45
Last modified on	2013-03-22 19:18:45
Owner	rspuzio (6075)
Last modified by	rspuzio (6075)
Numerical id	8
Author	rspuzio (6075)
Entry type	Algorithm
Classification	msc 65B99

From time to time, there arise occasions where one wants to raise a number to a large integer power. While one can compute x^n by multiplying x by itself n times, not only does this entail a large expenditure of effort when n is large, but roundoff errors can accumulate. A more efficient approach is to first rise x to powers of two by successive squaring, then multiply to obtain x^n. This procedure will now be explained by computing 1.08145^{187}.

The first step is to express n as a sum of powers of two, i.e. in base 2. In our example, we have $187 = 1 + 2 + 8 + 16 + 32 + 128$.

By the basic properties of exponentiation, we therefore have $x^{187} = x \cdot x^2 \cdot x^8 \cdot x^{16} \cdot x^{32} \cdot x^{128}$.

The next step is to raise 1.08145 to powers of two, which we accomplish by repeatedly squaring like so: (In order to guard against roundoff error, we work with more decimal places than needed and round off at the end of the computation.)

$$\begin{aligned}
x &= 1.08145 \\
x^2 &= 1.1695341 \\
x^4 &= (x^2)^2 = 1.1695341^2 = 1.3678100 \\
x^8 &= (x^4)^2 = 1.3678100^2 = 1.8709042 \\
x^{16} &= (x^8)^2 = 1.8709042^2 = 3.5002825 \\
x^{32} &= (x^{16})^2 = 3.5002825^2 = 12.251978 \\
x^{64} &= (x^{32})^2 = 12.251978^2 = 150.110965 \\
x^{128} &= (x^{64})^2 = 150.110965^2 = 26523642.95
\end{aligned}$$

Finally, we multiply together the appropriate powers to obtain our answer:

$$1.08145 \cdot 1.1695341 \cdot 1.8709042 \cdot 3.5002825 \cdot 12.251978 \cdot 26523642.95 = 2691617615$$

Hence, we compute $1.08145^{187} = 2.69162 \times 10^9$.

Note that this computation involved only 12 multiplications as opposed to 186 multiplications. More generally, we will require $\log_2 n$ repeated squarings and up to $\log_2 n$ multiplications of the repeated squares, or a total of between $\log_2 n$ and $2 \log_2 n$ multiplications to compute x^n this way.

More generally, the same method can be used to compute x^n when x is a complex number or a matrix, or something even more general, such as an element of a group. Thus this approach can be adapted to computing exponentials and trigonometric functions, numerical approximation of differential equation, and the like.