Addition-chain exponentiation

From Wikipedia, the free encyclopedia

In mathematics and computer science, optimal **addition-chain exponentiation** is a method of exponentiation by positive integer powers that requires a minimal number of multiplications. It works by creating the shortest addition chain that generates the desired exponent. Each exponentiation in the chain can be evaluated by multiplying two of the earlier exponentiation results. More generally, *addition-chain exponentiation* may also refer to exponentiation by non-minimal addition chains constructed by a variety of algorithms (since a shortest addition chain is very difficult to find).

The shortest addition-chain algorithm requires no more multiplications than binary exponentiation and usually less. The first example of where it does better is for a^{15}, where the binary method needs six multiplications but a shortest addition chain requires only five:

$a^{15} = a \times (a \times [a \times a^2]^2)^2$ (binary, 6 multiplications)
$a^{15} = a^3 \times ([a^3]^2)^2$ (shortest addition chain, 5 multiplications).

Table demonstrating how to do *Exponentiation* using *Addition Chains*

Number of Multiplications	Actual Exponentiation	Specific implementation of *Addition Chains* to do Exponentiation
0	a^1	a
1	a^2	a × a
2	a^3	a × a × a
2	a^4	(a × a→b) × b
3	a^5	(a × a→b) × b × a
3	a^6	(a × a→b) × b × b
4	a^7	(a × a→b) × b × b × a
3	a^8	((a × a→b) × b→d) × d
4	a^9	(a × a × a→c) × c × c
4	a^{10}	((a × a→b) × b→d) × d × b
5	a^{11}	((a × a→b) × b→d) × d × b × a
4	a^{12}	((a × a→b) × b→d) × d × d
5	a^{13}	((a × a→b) × b→d) × d × d × a
5	a^{14}	((a × a→b) × b→d) × d × d × b
5	a^{15}	((a × a→b) × b × a→e) × e × e
4	a^{16}	(((a × a→b) × b→d) × d→h) × h

On the other hand, the determination of a shortest addition chain is hard: no efficient optimal methods are currently known for arbitrary exponents, and the related problem of finding a shortest addition chain for a given set of exponents has been proven NP-complete.[1] Even given a shortest chain, addition-chain exponentiation requires more memory than the binary method, because it must potentially store many previous exponents from the chain. So in practice, shortest addition-chain exponentiation is primarily used for small fixed exponents for which a shortest chain can be precomputed and is not too large.

There are also several methods to *approximate* a shortest addition chain, and which often require fewer multiplications than binary exponentiation; binary exponentiation itself is a suboptimal addition-chain algorithm. The optimal algorithm choice depends on the context (such as the relative cost of the multiplication and the number of times a given exponent is re-used).[2]

The problem of finding the shortest addition chain cannot be solved by dynamic programming, because it does not satisfy the assumption of optimal substructure. That is, it is not sufficient to decompose the power into smaller powers, each of which is computed minimally, since the addition chains for the smaller powers may be related (to share computations). For example, in the shortest addition chain for a^{15} above, the subproblem for

Wikipedia The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Add links

a^6 must be computed as $(a^3)^2$ since a^3 is re-used (as opposed to, say, $a^6 = a^2(a^2)^2$, which also requires three multiplies).

Addition-subtraction–chain exponentiation [edit]

If both multiplication and division are allowed, then an addition-subtraction chain may be used to obtain even fewer total multiplications+divisions (where subtraction corresponds to division). However, the slowness of division compared to multiplication makes this technique unattractive in general. For exponentiation to negative integer powers, on the other hand, since one division is required anyway, an addition-subtraction chain is often beneficial. One such example is a^{-31}, where computing $1/a^{31}$ by a shortest addition chain for a^{31} requires 7 multiplications and one division, whereas the shortest addition-subtraction chain requires 5 multiplications and one division:

$a^{-31} = a/((((a^2)^2)^2)^2)^2$ (addition-subtraction chain, 5 mults + 1 div).

For exponentiation on elliptic curves, the inverse of a point (x, y) is available at no cost, since it is simply $(x, -y)$, and therefore addition-subtraction chains are optimal in this context even for positive integer exponents.[3]

References [edit]

1. ^ Downey, Peter; Leong, Benton; Sethi, Ravi (1981). "Computing sequences with addition chains". *SIAM Journal on Computing* **10** (3): 638–646. doi:10.1137/0210047.
2. ^ Gordon, D. M. (1998). "A survey of fast exponentiation methods" (PDF). *J. Algorithms* **27**: 129–146. doi:10.1006/jagm.1997.0913.
3. ^ François Morain and Jorge Olivos, "Speeding up the computations on an elliptic curve using addition-subtraction chains," *RAIRO Informatique théoretique et application* **24**, pp. 531-543 (1990).

- Donald E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd edition, §4.6.3 (Addison-Wesley: San Francisco, 1998).
- Daniel J. Bernstein, "Pippenger's Algorithm," to be incorporated into author's *High-speed cryptography* book. (2002)

Categories: Addition chains | Exponentials | Computer arithmetic algorithms