MATHEMATICS OF COMPUTATION, VOLUME 31, NUMBER 137
JANUARY 1977, PAGES 251–256

The Optimal Algorithm to Evaluate x^n Using Elementary Multiplication Methods

By D. P. McCarthy

Abstract. The optimality of the binary algorithm to evaluate x^n is established where x is an integer or a completely dense polynomial modulo m, n is a positive integer, and the multiplications are done using a simple improvement on the naive algorithm.

Introduction. The problem of finding the cheapest way to evaluate x^n has been considered by D. E. Knuth in [1] where, subject to the assumption that the cost of multiplying x^i by x^j is independent of x, i or j, he shows that the problem reduces to that of finding the shortest addition chain for n and discusses this problem at length. There is no known simple solution, but several algorithms are described that generate chains that are reasonably close to the shortest for moderate values of n.

W. M. Gentleman made a further contribution to the subject in [2] by showing that if x is a sparse polynomial and n is large enough then the cheapest way to evaluate x^n will eventually be by repeated multiplication by x. The difference between Knuth's and Gentleman's results is due to the fact that if i and j are large and x is sparse with $n + 1$ terms, then the multiplication of $x^i \cdot x^j$ has a cost in proportion to $(ij)^n$.

The purpose of this paper is to examine the particular case when the cost of multiplying x^i by x^j is proportional to ij; such a model applies to integers and dense polynomials modulo m.

Cost of Multiplication. We evaluate the cost of multiplying x^i by x^j using the principles outlined in [1], namely an enumeration of the number of primitive operations to be done.

We observe first of all that the only difference between integer and completely dense polynomial multiplication modulo m is that the carry is omitted in the latter case. If x is a p digit number then x^2 has $2p$ digits and x^i has ip digits, and the cost of multiplying x^i by x^j using the elementary algorithm is proportional to the number of digit by digit multiplications, that is ijp^2. If however $i = j$, then there is complete symmetry in the two arguments and only half the digit-by-digit multiplications are required, that is, $\frac{1}{2}ijp^2$.

If $C(e_1, \ldots, e_r; h_1, \ldots, h_s)$ is the cost of evaluating expressions e_1 to e_r given the expressions h_1 to h_s and if we normalize p^2 to 1, then we may write

Received May 12, 1975; revised May 3, 1976.
AMS (MOS) subject classifications (1970). Primary 68A20.
Key words and phrases. Symbolic algebraic manipulation, computational complexity, optimal multiplication chains.

$$C(x^i \cdot x^j : x^i, x^j) = \begin{pmatrix} 2ij, & i \neq j \\ i^2, & i = j \end{pmatrix}. \tag{1}$$

The Binary Algorithm. The binary representation of the number n may be used to generate short addition chains for n and hence an evaluation of x^n needing relatively few multiplications; an algorithm exploiting this is given in [1, p. 399] and called the binary algorithm and the example $n = 15$ used to prove that the algorithm does not lead to the shortest multiplication chain for x^{15}. The following recursive version of the algorithm assumes a special squaring algorithm $\mathrm{sq}(x) = x^2$ which avails of the symmetry to reduce by half the number of digit-by-digit multiplications required. It has the further advantage of requiring a single right to left scan of the binary representation of n.

$$\mathrm{bexp}(x, n) = \begin{pmatrix} x, & n = 1 \\ \mathrm{sq}\left(\mathrm{bexp}\left(x, \frac{n}{2}\right)\right), & n \text{ even} \\ x \cdot \mathrm{sq}\left(\mathrm{bexp}\left(x, \frac{n-1}{2}\right)\right), & n \text{ odd} \end{pmatrix}. \tag{2}$$

Notation. We define $C_b(x^n) = C(\mathrm{bexp}(x, n): x)$ and $\mathbf{C}(e_1, \ldots, e_r: h_1, \ldots, h_s)$ as the cost of the cheapest possible evaluation of $e_1, \ldots, e_r$ given $h_1, \ldots, h_s$. In particular, $\mathbf{C}(x^n) = \mathbf{C}(x^n: x)$ is the cost of the cheapest possible evaluation of x^n using the multiplication algorithm outlined above. Finally, the sequence $1 = b_0^n, \ldots, b_s^n = n$ is the addition chain defined by the binary algorithm for n.

THEOREM. *For all $n > 0$ and x such that* (1) *above holds we have*

$$\mathbf{C}(x^n) = C_b(x^n) \tag{3}$$

and $b_0^n, \ldots, b_s^n$ is uniquely the cheapest chain for x^n.

To prove this we need the result that for all $0 < m < n$,

$$\mathbf{C}(x^n) - \mathbf{C}(x^{n-m}) \leqslant 2mn - m(m+1). \tag{4}$$

Since

$$x^n = \underbrace{x \cdot x \cdot \cdots \cdot x}_{m \text{ times}} \cdot x^{n-m},$$

then

$$\mathbf{C}(x^n) \leqslant \mathbf{C}(x^{n-m}) + 2\{1(n-m) + 1(n-m+1) + \cdots + 1(n-1)\}$$

and hence the result (4).

Next it may readily be shown that $\mathbf{C}(x^n) = C_b(x^n)$ for $n = 1, 2, 3, 4$ and that the addition chain defined by $\mathrm{bexp}(x, n)$ is uniquely the cheapest; therefore, we commence by assuming

$$\mathbf{C}(x^i) = C_b(x^i), \quad 1 \leqslant i \leqslant p-1, \tag{5}$$

and

$$1 = b_0^i, \ldots, b_s^i = i \text{ is unique for } 1 \leqslant i \leqslant p-1.$$

Next we deny the theorem by assuming that there exists an addition chain $a_0, \ldots, a_r \neq b_0^p, \ldots, b_s^p$ which is cheaper or as cheap as that defined by bexp(x, n). That is if $C_a(x^p)$ is the cost of evaluating x^p using the chain A, we have

$$C_a(x^p) = \mathbf{C}(x^p) \leqslant C_b(x^p); \tag{6}$$

and clearly, we must have $a_{r-1} \neq b_{s-1}^p$ otherwise A would point to a cheaper method to evaluate $x^{a_{r-1}}$ or a nonunique chain for $x^{a_{r-1}}$ in contradiction of (5). We have then

$$\mathbf{C}(x^p) = \mathbf{C}(x^{a_{r-1}} \cdot x^{p-a_{r-1}} : x^{a_{r-1}}, x^{p-a_{r-1}}) + \mathbf{C}(x^{a_{r-1}}, x^{p-a_{r-1}} : x);$$

and since for all possible a_{r-1}, $p - a_{r-1} < a_{r-1}$,

$$\mathbf{C}(x^p) \geqslant \mathbf{C}(x^{a_{r-1}} \cdot x^{p-a_{r-1}} : x^{a_{r-1}}, x^{p-a_{r-1}}) + \mathbf{C}(x^{a_{r-1}} : x);$$

and this with (5) and (6) implies

$$C_b(x^p) \geqslant \mathbf{C}(x^{a_{r-1}} \cdot x^{p-a_{r-1}} : x^{a_{r-1}}, x^{p-a_{r-1}}) + C_b(x^{a_{r-1}}). \tag{7}$$

We now consider separately the four possibilities that p and a_{r-1} be, respectively, odd or even.

Let $p = 2n + a$ and $p - a_{r-1} = 2m + b$ where $n > m > 0$ and $a, b \in \{0, 1\}$ so that $a_{r-1} = 2(n-m) + (a-b) = 2j + c$ where $j > 0$ and $c \in \{-1, 0, +1\}$ whereupon

$$C_b(x^p) = C_b(x^{2n+a}) = n^2 + 4an + C_b(x^n), \tag{8}$$

$$\mathbf{C}(x^{a_{r-1}} \cdot x^{p-a_{r-1}} : x^{a_{r-1}}, x^{p-a_{r-1}}) = 2(2j + c)(2m + b), \tag{9}$$

$$C_b(x^{a_{r-1}}) = C_b(x^{2j+c}) = \begin{pmatrix} j^2 + C_b(x^j), & c = 0 \\ j^2 + 4j + C_b(x^j), & c = 1 \\ (j-1)^2 + 4(j-1) + C_b(x^{j-1}), & c = -1 \end{pmatrix}.$$

By (4) $C_b(x^j) \leqslant C_b(x^{j-1}) + 2(j-1)$ so

$$C_b(x^{a_{r-1}}) \geqslant \begin{pmatrix} j^2 + C_b(x^j), & c = 0 \\ j^2 + 4j + C_b(x^j), & c = 1 \\ j^2 - 1 + C_b(x^j), & c = -1 \end{pmatrix},$$

$$C_b(x^{a_{r-1}}) \geqslant j^2 + C_b(x^j) + f(c), \tag{10}$$

where $f(0) = 0, f(1) = 4j, f(-1) = -1$.

Furthermore, for all $a, b \in \{0, 1\}$ we have

$$0 < m < \frac{2n + a - 2b}{4} \tag{11}$$

which implies at least $0 < m \leqslant n/2$.

We substitute (8), (9), (10) into (7) to obtain

$$n^2 + 4an + C_b(x^n) \geqslant 2(2j + c)(2m + b) + j^2 + C_b(x^j) + f(c)$$

so

$$C_b(x^n) - C_b(x^j) \geqslant 2(2j + c)(2m + b) + j^2 + f(c) - n^2 - 4an.$$

By (4),

$$2mn - m^2 - m \geqslant 2(2j + c)(2m + b) + j^2 + f(c) - n^2 - 4an.$$

Then letting $j = n - m$, $c = a - b$ and regrouping we get

$$6m^2 + (8b - 4a - 1)m - 2bc - f(c) \geqslant 4mn + 4(b - a)n. \tag{12}$$

We next evaluate (12) for all $a, b \in \{0, 1\}$ to examine the four cases.

(1) $a = b = 0$ implies $c = 0$ and $f(0) = 0$ yielding

$$6m^2 - m \geqslant 4mn;$$

and since $m > 0$,

$$m \geqslant 2n/3 + 1/6. \tag{13}$$

(2) $a = 0, b = 1$ implies $c = -1$ and $f(-1) = -1$ yielding

$$6m^2 + 7m + 3 \geqslant 4mn + 4n,$$

but $9m > 7m$ for all $m > 0$ so that we get

$$6m^2 + 9m + 3 > 4n(m + 1), \quad 6m + 3 > 4n, \quad m > \frac{2n}{3} - \frac{1}{2}. \tag{14}$$

(3) $a = 1, b = 0$ implies $c = 1$ and $f(1) = 4n - 4m$ yielding

$$6m^2 - m \geqslant 4mn, \quad m \geqslant \frac{2n}{3} + \frac{1}{6}. \tag{15}$$

(4) $a = b = 1$ implies $c = 0$ and $f(0) = 0$ yielding

$$6m^2 + 3m \geqslant 4mn, \quad m \geqslant \frac{2n}{3} - \frac{1}{2}. \tag{16}$$

The results (13), (14), (15), (16) all contradict the constraint (11) and show exhaustively that the assumption that there exists another chain different from $b_0^p, \ldots, b_s^p = p$ such that $C_a(x^p) \leqslant C_b(x^p)$ leads to a contradiction about a_{r-1}. Thus we have $\mathbf{C}(x^p) = C_b(x^p)$ for all p, and the addition chain $b_0^p, \ldots, b_s^p$ defines uniquely the cheapest multiplication chain to generate x^p.

Completely Naive Multiplication. Since it does not always prove convenient to write a special squaring algorithm, it is important to consider the case where no

advantage is taken of the symmetry in the product $x^i \cdot x^i$ so that it has cost relative to (1) of

$$C(x^i \cdot x^j: x^i, x^j) = 2ij \quad \text{for all } i \,\&\, j;$$

and we shall use $\tilde{C}(x^n)$ to denote the cost of x^n evaluated in this way. Examination of the 69, 169 possible chains for x^n, $1 \leqslant n \leqslant 20$ shows that

$$\tilde{\mathbf{C}}(x^n) = C_b(x^n), \qquad 1 \leqslant n \leqslant 20, \tag{17}$$

but a proof for all n has yet to emerge. If, as seems probable, (17) is true for all n, it will still lack the uniqueness of (5) since

$$\tilde{C}(x(x^n \cdot x^n): x^n, x) = \tilde{C}(x^n(x^n \cdot x): x^n, x) = 2n^2 + 4n,$$

so that for any odd n there will be at least two different chains costing the same. An example will serve to highlight the contrast between the uniqueness of $C_b(x^{15}) = \mathbf{C}(x^{15})$ and the nonuniqueness of $\tilde{C}_b(x^{15}) = \tilde{\mathbf{C}}(x^{15})$ and also the shortest chains for $n = 15$.

bexp $(x, 15)$ yields the chain (1, 2, 3, 6, 7, 14, 15) with cost $\mathbf{C}(x^{15}) = 103$ and $\tilde{\mathbf{C}}(x^{15}) = 162$. However, the chains (1, 2, 3, 6, 7, 8, 15), (1, 2, 3, 4, 7, 14, 15), (1, 2, 3, 4, 7, 8, 15) also all have the cost $\tilde{C}(x^{15}) = 162$ while the four shortest possible chains for 15, $S_1 = (1, 2, 3, 5, 10, 15)$, $S_2 = (1, 2, 3, 6, 9, 15)$, $S_3 = (1, 2, 3, 6, 12, 15)$, $S_4 = (1, 2, 4, 5, 10, 15)$ have cost $C(S_1) = 142$, $C(S_2) = 158$, $C(S_3) = 122$, $C(S_4) = 138$ and $\tilde{C}(S_1) = \tilde{C}(S_2) = \tilde{C}(S_3) = \tilde{C}(S_4) = 168$. It is seen that though requiring one less multiplication the cheapest of the shortest chains is approximately 20% dearer than the cheapest chain.

Incidental Results. (1) *Bounds on* $\mathbf{C}(x^n)$. Examination of the case $n = 2^i$ shows

$$\mathbf{C}(x^n) = \frac{n^2 - 1}{3}, \tag{18}$$

whereas $n = 2^i - 1$ yields

$$\mathbf{C}(x^n) = \frac{n^2 + 8n - 9\log_2(n+1)}{3}. \tag{19}$$

Since these represent the best and worst cases, we have bounds on $\mathbf{C}(x^n)$

$$\frac{n^2 - 1}{3} \leqslant \mathbf{C}(x^n) \leqslant \frac{n^2 + 8n - 9\log_2(n+1)}{3}.$$

From this we may observe that, assuming it costs $2n$ to divide x^n by x, it will never be cheaper to obtain x^{2i-1} by dividing x^{2i} by x since

$$C(x^n \div x) - \mathbf{C}(x^{n-1}) = \frac{n^2 - 1}{3} + 2n - \frac{(n-1)^2 + 8(n-1) - 9\log_2 n}{3}$$

$$= 3\log_2 n + 2 > 0 \quad \text{for all } n = 2^i.$$

(2) *Comparison with Repeated Multiplication by* x. We may contrast the bounds (18) and (19) with $C^*(x^n)$, the cost of evaluation of x^n by repeated multiplication by x. We have

$$C^*(x^n) = 1 \cdot 1 + 2 \sum_{j=2}^{n-1} j \text{ for all } n > 1$$
$$= n^2 - n - 1$$

so for large n both $C^*(x^n)$ and $\mathbf{C}(x^n)$ are both $O(n^2)$.

(3) *Space Requirements.* Lastly, we note that if the squaring and multiplication algorithms do not destroy their arguments until completion of the operation and they can doubly reference their arguments, then the space required by the binary algorithm to evaluate x^n, denoted by $S_b(x^n)$, is given by

$$S_b(x^n) = \begin{pmatrix} \frac{3n}{2}, & n \text{ even} \\ 2n, & n \text{ odd} \end{pmatrix},$$

and this is clearly minimal.

Conclusion. The importance of the above theorem is that it assures us that, for integer and dense polynomial multiplication modulo m using a naive algorithm, the simple binary algorithm is the best one to use despite the fact that it sometimes takes more multiplication steps than other algorithms.

Department of Computer Science
Trinity College
Dublin 2, Ireland

1. D. E. KNUTH, *The Art of Computer Programming*, Vol. 2, Addison-Wesley, Reading, Mass., 1969. MR **44** # 3531.

2. W. M. GENTLEMAN, "Optimal multiplication chains for computing a power of a symbolic polynomial," *Math. Comp.*, v. 26, 1972, pp. 935–939. MR **47** # 2855.