JOURNAL OF ALGORITHMS 6, 376–380 (1985)

On the Complexity of Calculating Factorials

PETER B. BORWEIN*

Department of Mathematics, Statistics, and Computing Science, Dalhousie University, Halifax, Nova Scotia, Canada B3H 4H8

Received August 18, 1983

It is shown that $n!$ can be evaluated with time complexity $O(\log\log n\, M(n\log n))$, where $M(n)$ is the complexity of multiplying two n-digit numbers together. This is effected, in part, by writing $n!$ in terms of its prime factors. In conjunction with a fast multiplication this yields an $O(n(\log n \log\log n)^2)$ complexity algorithm for $n!$. This might be compared to computing $n!$ by multiplying 1 times 2 times 3, etc., which is $\Omega(n^2\log n)$ and also to computing $n!$ by binary splitting which is $O(\log n\, M(n\log n))$. © 1985 Academic Press, Inc.

From a complexity point of view one of the least efficient ways to calculate large factorials is to multiply the successive integers together. Any binary splitting on the multiplication substantially reduces the complexity. However, the best available asymptotic bounds are attained by writing $n!$ in terms of its prime divisors.

We let $M(n)$ denote the time complexity of multiplying two n (decimal) digit numbers together and we make the assumption that M is non-decreasing and $M(2n) \leqslant 4M(n) \leqslant 2M(2n)$. This is satisfied by usual multiplication which is $O(n^2)$ and Schönhage–Strassen multiplication which is $O(n\log n\log\log n)$.

Suppose we evaluate $n!$ by multiplying 1 times 2 times 3, etc. This calculation requires computing $k!$ for each $k \leqslant n$ and since $k!$ has $\Omega(k\log k)$ digits we observe that this computation is

$$\Omega\left(\sum_{k=1}^{n} k\log k\right) = \Omega\left(n^2\log n\right).$$

That is, $cn^2\log n$ is a lower bound for the time of this calculation.

*Research supported in part by NSERC of Canada.

0196-6774/85 $3.00

It is straightforward to reduce the time to $O(\log n\, M(n \log n))$ as the first proposition shows. With a fast multiplication this provides an $O(n(\log n)^3 \log\log n)$ method for computing factorials.

PROPOSITION 1. *The time required to multiply k α-digit integers together is $O(\log k\, M(k \cdot \alpha))$.*

Proof. Suppose $k = 2^h$. We divide the k numbers into pairs and multiply to construct 2^{h-1} 2α-digit numbers. Continuing inductively we see that the complexity of the calculation at the ith step is $\leqslant 2^{h-i}M(2^{i-1}\alpha)$ and thus, the total complexity is

$$\leqslant \sum_{i=1}^{h} 2^{h-i}M(2^{i-1}\alpha)$$

$$\leqslant \sum_{i=1}^{h} 2^{h-i}2^{-(h-i+1)}M(2^h\alpha)$$

$$= \frac{h}{2}M(2^h\alpha). \qquad \square$$

This is only an improvement provided a fast multiplication is used. With an ordinary multiplication both the above method and the usual method for evaluating $n!$ are of complexity $O(n^2(\log n)^2)$.

The order of the complexity estimate for computing $n!$ as in Proposition 1 is independent of how the numbers 1 through n are grouped initially

complexity by taking advantage of the fact that some of the multiplications are of lower order.

We now show how to derive a complexity of $O(\log\log n\, M(n \log n))$ for $n!$. The steps of the calculation are as follows:

Step 1. Construct a table of primes $p, 2 \leqslant p \leqslant n$.

Step 2. Compute the exponent of each p in the factorization of $n!$.

Step 3. Calculate the $O(\log n)$ numbers

$$\alpha_i = (\Pi p)^{2^i},$$

where the product is taken over those primes p whose index in $n!$ has a non-zero multiple of 2^i in its base two expansion.

Step 4. Compute $n! = \Pi\alpha_i$.

PROPOSITION 2. *The complexity of evaluating $n!$, using steps 1–4, is*

$$O(\log\log n\, M(n \log n)).$$

Proof. *The complexity of Step 1.* We use a straightforward sieve method to construct a table of primes. Start with a table of integers 2 through n. At the ith stage we remove all integers that are multiples of p_i the ith prime. The complexity of this ith operation is $O((n/p_i)\log n)$, since it requires n/p_i additions and comparisons of numbers of length at most $\log n$. (Note that the above O is in both n and i. We shall adopt the convention that $O(f(n,i))$ means that the bound is provided as $n+i\to\infty$ through the feasible values.) Since $\Sigma_{p_i\leqslant n}(1/p_i)=O(\log\log n)$, the total complexity is

$$O\Big(\sum_{p_i\leqslant n}(n/p_i)\log n\Big)=O(n\log n\log\log n). \tag{1}$$

The crude estimate, though sufficient for our purposes, is far from best possible. A bound of $O(n\log n/\log\log n)$ is established in [5].

The complexity of Step 2. The index of p in $n!$ is given by $\Sigma_{i=1}^{\infty}[n/p^i]$, where $[x]$ denotes the greatest integer less than or equal to x. This quantity can be evaluated employing $O(\log n)$ divisions and additions and hence has complexity

$$O(\log n\, M(\log n)).$$

Here we have used the fact that division is linearly equivalent to multiplication (see [1]). Thus, the complexity for working out the indices of all the primes $\leqslant n$ is

$$O(nM(\log n))\leqslant O(M(n\log n)), \tag{2}$$

since there are $O(n/\log n)$ primes less than n.

The complexity of Step 3. Consider β_i, where

$$\beta_i=(\alpha_i)^{1/2^i}=\prod p,$$

the product being taken over those primes p whose index in $n!$ has a non-zero multiple of 2^i (i.e., a non-zero ith digit) in its binary expansion. Each prime p in the above product satisfies

$$p\leqslant\frac{2n}{2^i}.$$

Note that the index of p is bounded above by

$$\sum_{i=1}^{\infty}\frac{n}{p^i}\leqslant\frac{2n}{p}.$$

Thus, since there are at most $O(2n/2^i\log(2n/2^i))$ primes less than $2n/2^i$,

we have

$$\beta_i \leqslant \left(\frac{2n}{2^i}\right)^{(c\cdot 2n/2^i \log(2n/2^i))}$$
$$= e^{c2n/2^i} = e^{c^*n/2^i}, \tag{3}$$

where c^* and c are independent of n and i. Also

$$\alpha_i \leqslant e^{c^*n}. \tag{4}$$

Now β_i is a product of $O(n/2^i \log(n/2^i))$ terms each of $O(\log(n/2^i))$ digits. It follows from Proposition 1 that the time required to compute β_i is

$$O\left(\log\frac{n}{2^i} M\left(\frac{n}{2^i}\right)\right). \tag{5}$$

The order in (5) is independent of both n and i. Thus, the complexity of computing all the β_i is

$$O\left(\sum_{i=1}^{O(\log n)} \log\frac{n}{2^i} M\left(\frac{n}{2^i}\right)\right) = O(\log n\, M(n)) = O(M(n \log n)). \tag{6}$$

If a is an α-digit integer then a^{2^k} can be calculated by repeated squaring in time

$$O\left(\sum_{h=0}^{k-1} M(2^h\alpha)\right) = O(M(2^k\alpha)).$$

Thus, by (3), given β_i we can compute $\alpha_i = \beta_i^{2^i}$ with complexity of

$$O\left(M\left(\frac{c^*n}{2^i}2^i\right)\right) = O(M(n))$$

and since there are $O(\log n)$ of the α_i they can all be computed in time

$$O(\log n\, M(n)) = O(M(n \log n)). \tag{7}$$

The complexity of Step 4. Given the α_i we wish to calculate $\prod \alpha_i = n!$. This, by (4), requires $O(\log n)$ multiplications of $O(n)$ digit numbers and, by Proposition 1, is of complexity.

$$O(\log\log n\, M(n \log n)). \tag{8}$$

The total complexity is now given by (1), (2), (6), (7), and (8) which is dominated by (8) and is

$$O(\log\log n\, M(n \log n)). \qquad \square$$

The few pieces of number theory needed for Proposition 2 may be found in [2] and, in part, in [3]. A discussion of fast multiplication is available in [4]. Since $n!$ has $O(n \log n)$ digits it seems unlikely that it has lower complexity than $O(M(n \log n))$. Whether the $\log \log n$ is necessary is more problematical.

REFERENCES

1. R. P. BRENT, The complexity of multiple precision arithmetic, *in* "Complexity of Computational Problem Solving" (R. S. Anderssen and R. P. Brent, Eds.), pp. 126–165, Univ. of Queensland Press, Brisbane, 1976.
2. G. H. HARDY, AND E. M. WRIGHT, "An Introduction to the Theory of Numbers," Oxford Univ. Press, London/New York, 1960.
3. D. E. KNUTH, "The Art of Computer Programming: Fundamental Algorithms," 2nd ed., Vol. 1, Addison–Wesley, Reading, Mass., 1973.
4. D. E. KNUTH, "The Art of Computer Programming: Seminumerical Algorithms," 2nd ed., Vol. 2, Addison–Wesley, Reading, Mass., 1981.
5. P. A. PRITCHARD, Sublinear additive sieve for finding prime numbers, Comm. ACM **1** (1981), 18–23.