

用分治算法求大整数相乘问题的进一步分析

王念平, 金晨辉

(解放军信息工程大学电子技术学院, 河南郑州 450004)

摘 要: 对利用分治算法解决大整数相乘问题作了进一步深入的研究和分析. 在原来的分治算法的基础上, 将输入规模为 n 的两个大整数各分成规模相等的 $k(2 \leq k \leq n)$ 部分, 证明了通过恒等变形可将其乘积中的 k^2 次乘法降为 $k(k+1)/2$ 次; 给出了计算两个大整数乘积的计算复杂度; 证明了利用分治算法将两个大整数各分成规模相等的两部分来进行处理时的计算复杂度是最小的, 进而表明利用分治算法将大整数各分成规模相等的两部分来进行处理是合理的.

关键词: 大整数相乘问题; 分治算法; 计算复杂度

中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2008)01-0133-03

More Analyses for Multiplication of Large Integers Using the Algorithm of Divide and Conquer

WANG Niarr ping, JIN Cherr hui

(Institute of Electronic Technology, The PLA Information Engineering University, Zhengzhou, Henan 450004, China)

Abstract: More deep analyses for multiplication of large integers is given using the algorithm of divide and conquer. Based on the formerly algorithm of divide and conquer, we divide each of large integers into $k(2 \leq k \leq n)$ parts which are the same size. It is proven that we can reduce the times of multiplication from k^2 to $k(k+1)/2$ in the product of two integers. The complexity for calculating the product of two large integers is given and is proven to be the lowest when we divide each of large integers into two parts which are the same size. Furthermore, it is shown to be reasonable to divide each of two large integers into two parts using the algorithm of divide and conquer.

Key words: multiplication of large integers; the algorithm of divide and conquer; complexity

1 引言

对于输入规模为 n 的两个大整数相乘, 按照普通乘法显然需要 $O(n^2)$ 次“位”运算. 如果将输入的两个大整数各分成规模相等的两部分, 按照分治算法, 只需 $O(n^{\lg_2 3}) \approx O(n^{1.59})$ 次“位”运算^[1~3].

具体地, 设 X 和 Y 是两个 n 位二进制整数. 为方便, 不妨假设 n 为 2 的方幂, $n = 2^l (l \geq 1)$. 把 X 和 Y 各分成规模相等的两部分, $X = A_1 2^{n/2} + A_0$, $Y = B_1 2^{n/2} + B_0$. 这里 A_1, A_0, B_1, B_0 都有 $n/2$ 位. 显然 $XY = A_1 B_1 2^n + (A_1 B_0 + B_1 A_0) 2^{n/2} + A_0 B_0$. 这样, 求 X 和 Y 的乘积就转化为 4 个子问题: $A_1 B_1, A_1 B_0, B_1 A_0$ 和 $A_0 B_0$. 此外, 一些加法和移位的次数不超过 cn (c 为某一常数, 下同). 故若设两个 n 位二进制整数相乘的计算复杂度为 $T(n)$ (这里把一次位运算当作“基本操作”, 下同), 则可得到以下递归式

$$T(n) = \begin{cases} 1 & n = 1 \\ 4T(n/2) + cn & n > 1 \end{cases}$$

从而 $T(n) = O(n^2)$. 这样做计算复杂度并没有得到改进, 这是因为这四个子问题并不是独立的! 事实上, $A_1 B_0 + B_1 A_0 = (A_1 + A_0)(B_1 + B_0) - A_1 B_1 - A_0 B_0$. 从而 $XY = A_1 B_1 2^n + [(A_1 + A_0)(B_1 + B_0) - A_1 B_1 - A_0 B_0] \times 2^{n/2} + A_0 B_0$, 即通过恒等变形可把四次乘法降为三次: $A_1 B_1, (A_1 + A_0)(B_1 + B_0)$ 和 $A_0 B_0$. 于是

$$T(n) = \begin{cases} 1, & n = 1 \\ 3T(n/2) + cn, & n > 1 \end{cases}$$

故 $T(n) = O(n^{\lg_2 3}) \approx O(n^{1.59})$.

但能否进一步将两个大整数各分成规模相等的三部分、四部分甚至更多部分, 从而再次改进其计算复杂度呢? 当输入的规模 n 很大时, 这样的改进是有意义的. 在已有的文献中都是将大整数各分成规模相等的两部分来进行处理, 但缺乏对这个问题的进一步分析. 本文经过深入研究表明: 大整数相乘时, 若将输入规模为 n 的两个大整数各分成规模相等的 $k(2 \leq k \leq n)$ 部分, 并按照与 $k=2$ 时类似的方法进行处理, 则计算两个大整数的乘积时以 $k=2$ 时的计算复杂度为最小, 从而利

用分治算法将大整数各分成规模相等的两部分来进行处理是合理的.

2 有关的引理

引理 1^[4] 数列 $\left\{a_u = \left[1 + \frac{1}{u}\right]^u\right\}_{u=0}^{u=\infty}$ 是单调递增

的, 且对任一正整数 u , $a_u = \left[1 + \frac{1}{u}\right]^u < e$, 这里 e 表示自然对数的底数.

引理 2 数列 $\left\{a_u = \frac{2^{u+1}}{u+2}\right\}_{u=0}^{u=\infty}$ 是单调递增的.

引理 3 设 $A_i, B_i (0 \leq i \leq m, m \text{ 为某一正整数})$ 都是正整数, 则

$$A_m B_{m-1} + A_{m-1} B_m = \left[\sum_{j=m-i}^m A_j \right] \left[\sum_{j=m-i}^m B_j \right] - \left[\sum_{j=m-i}^{m-1} A_j \right] \left[\sum_{j=m-i}^{m-1} B_j \right] - \sum_{j=m-i+1}^{m-1} (A_m B_j + B_m A_j) - A_n B_m$$

引理 4 函数 $f(x) = \log_x [(x+1)/2]$ 在区间 $(3, +\infty)$ 上是严格单调递增的.

3 大整数相乘问题的进一步分析

在以下的分析中, 为方便, 仍然假设 n 为 $k(2 \leq k \leq n)$ 的方幂, $n = k^l$. 将输入规模为 n 的两个二进制整数 X 和 Y 各分成规模相等的 $k(1 \leq k \leq n)$ 部分, 并记

$$X = A_{k-1} 2^{(k-1)n/k} + A_{k-2} 2^{(k-2)n/k} + \dots + A_1 2^{n/k} + A_0$$

$$Y = B_{k-1} 2^{(k-1)n/k} + B_{k-2} 2^{(k-2)n/k} + \dots + B_1 2^{n/k} + B_0$$

按照普通乘法, 计算 X 和 Y 的乘积可得到 k^2 次规模为 n/k 位的二进制整数乘法, 且这 k^2 次乘法可统一表示为 $A_i B_j, 0 \leq i, j \leq k-1$.

定理 1 若将输入规模为 n 的两个二进制整数 X 和 Y 各分成规模相等的 $k(2 \leq k \leq n)$ 部分, 则通过恒等变形可以把 k^2 次规模为 n/k 位的二进制整数乘法降为 $k(k+1)/2$ 次, 且这 $k(k+1)/2$ 次乘法可统一表示为

$$\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right], 0 \leq j \leq k-1, 0 \leq t \leq j$$

证明: (数学归纳法)

(1) $k=2$ 时, 由引言中的分析知, 定理结论成立.

(2) 假设 $k=m(2 \leq m \leq n-1)$ 时定理结论成立, 则 $k=m+1$ 时, 将 X 和 Y 各分成规模相等的 $m+1$ 部分 (这里不妨假设 $n=(m+1)^l$), 此时

$$X = A_m 2^{mn/(m+1)} + A_{m-1} 2^{(m-1)n/(m+1)} + \dots + A_1 2^{n/(m+1)} + A_0$$

$$Y = B_m 2^{mn/(m+1)} + B_{m-1} 2^{(m-1)n/(m+1)} + \dots + B_1 2^{n/(m+1)} + B_0$$

按照普通乘法计算 X 和 Y 的乘积, 显然需要 $(m+1)^2$ 次规模为 $n/(m+1)$ 位的二进制整数乘法. 令

$$S_1 = A_{m-1} 2^{(m-1)n/(m+1)} + \dots + A_1 2^{n/(m+1)} + A_0$$

$$S_2 = B_{m-1} 2^{(m-1)n/(m+1)} + \dots + B_1 2^{n/(m+1)} + B_0$$

则 © 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$\begin{aligned} XY &= (A_m 2^{mn/(m+1)} + A_{m-1} 2^{(m-1)n/(m+1)} + \dots + A_1 2^{n/(m+1)} + A_0) \\ &\quad \times (B_m 2^{mn/(m+1)} + B_{m-1} 2^{(m-1)n/(m+1)} + \dots + B_1 2^{n/(m+1)} + B_0) \\ &= A_m B_m 2^{2mn/(m+1)} + A_m 2^{mn/(m+1)} S_2 + B_m 2^{mn/(m+1)} S_1 + S_1 S_2 \\ &= A_m B_m 2^{2mn/(m+1)} + [(A_m B_{m-1} + A_{m-1} B_m) 2^{(2m-1)n/(m+1)} \\ &\quad + (A_m B_{m-2} + A_{m-2} B_m) 2^{(2m-2)n/(m+1)} + \dots + (A_m B_0 + A_0 B_m) 2^{mn/(m+1)}] + S_1 S_2 \end{aligned}$$

由归纳假设, 对于乘积项 $S_1 S_2$, 可通过恒等变形把 m^2 次规模为 $n/(m+1)$ 位的二进制整数乘法降为 $m(m+1)/2$ 次, 且这 $m(m+1)/2$ 次乘法可统一表示为 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right] (0 \leq j \leq m-1, 0 \leq t \leq j)$.

对于项 $A_m 2^{mn/(m+1)} S_2 + B_m 2^{mn/(m+1)} S_1$, 我们以下证明: $A_m 2^{mn/(m+1)} S_2 + B_m 2^{mn/(m+1)} S_1$ 通过恒等变形所产生的乘法与 $S_1 S_2$ 通过恒等变形所产生的乘法 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right] (0 \leq j \leq m-1, 0 \leq t \leq j)$ 相比, 仅仅新增加了 $m+1$ 次乘法, 且这 $m+1$ 次乘法为

$$A_m B_m, (A_m + A_{m-1})(B_m + B_{m-1}), \dots, (A_m + A_{m-1} + \dots + A_1)(B_m + B_{m-1} + \dots + B_1), (A_m + A_{m-1} + \dots + A_0)(B_m + B_{m-1} + \dots + B_0)$$

证明过程如下:

(a) 由恒等式

$$A_m B_{m-1} + A_{m-1} B_m = (A_m + A_{m-1})(B_m + B_{m-1}) - A_{m-1} B_{m-1} - A_m B_m$$

知, $A_m B_{m-1} + A_{m-1} B_m$ 通过恒等变形所产生的乘法与 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right] (0 \leq j \leq m-1, 0 \leq t \leq j)$ 相比新增加了两次乘法: $(A_m + A_{m-1})(B_m + B_{m-1})$ 和 $A_m B_m$.

(b) 由恒等式

$$A_m B_{m-2} + A_{m-2} B_m = (A_m + A_{m-1} + A_{m-2})(B_m + B_{m-1} + B_{m-2}) - (A_{m-1} + A_{m-2})(B_{m-1} + B_{m-2}) - (A_m B_{m-1} + A_{m-1} B_m) - A_m B_m$$

知, $A_m B_{m-2} + A_{m-2} B_m$ 通过恒等变形所产生的乘法与 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right] (0 \leq j \leq m-1, 0 \leq t \leq j)$ 相比新增加了三次乘法: $(A_m + A_{m-1} + A_{m-2})(B_m + B_{m-1} + B_{m-2})$ 、 $A_m B_{m-1} + A_{m-1} B_m$ 和 $A_m B_m$, 但 $A_m B_{m-1} + A_{m-1} B_m$ 可归结为 (a) 中的恒等式, 从而 $A_m B_{m-2} + A_{m-2} B_m$ 通过恒等变形最终所产生的乘法与 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right] (0 \leq j \leq m-1, 0 \leq t \leq j)$ 相比新增加了三次乘法: $(A_m + A_{m-1} + A_{m-2})(B_m + B_{m-1} + B_{m-2})$ 、 $(A_m + A_{m-1})(B_m + B_{m-1})$ 和 $A_m B_m$.

依次类推, 对任意的 $i, 3 \leq i \leq m$ 由引理 3 中的恒等式知, $A_m B_{m-i} + A_{m-i} B_m$ 通过恒等变形最终所产生的

乘法与 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right]$ ($0 \leq j \leq m-1, 0 \leq t \leq j$) 相比新增加了 $i+1$ 次乘法: $\left[\sum_{j=t}^m A_j \right] \left[\sum_{i=t}^m B_j \right]$ ($m-i \leq t \leq m-1$) 和 $A_m B_m$. 故

$$A_m 2^{mn/(m+1)} S_2 + B_m 2^{mn/(m+1)} S_1 = (A_m B_{m-1} + A_{m-1} B_m) 2^{(2m-1)n/(m+1)} + (A_m B_{m-2} + A_{m-2} B_m) 2^{(2m-2)n/(m+1)} + \dots + (A_m B_0 + A_0 B_m) 2^{mn/(m+1)}$$

通过恒等变形最终所产生的乘法与 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right]$ ($0 \leq j \leq m-1, 0 \leq t \leq j$) 相比新增加了 $m+1$ 次乘法, 这 $m+1$ 次乘法为:

$$A_m B_m, (A_m + A_{m-1})(B_m + B_{m-1}), \dots, (A_m + A_{m-1} + \dots + A_1)(B_m + B_{m-1} + \dots + B_1), (A_m + A_{m-1} + \dots + A_0)(B_m + B_{m-1} + \dots + B_0)$$

从而

$$XY = A_m B_m 2^{2mn/(m+1)} + A_m 2^{mn/(m+1)} S_2 + B_m 2^{mn/(m+1)} S_1 + S_1 S_2$$

通过恒等变形最终所产生的乘法次数一共有 $m(m+1)/2 + (m+1) = (m+1)(m+2)/2$ 次. 再由归纳假设知, $S_1 S_2$ 通过恒等变形所产生的 $m(m+1)/2$ 次乘法可

表示为 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right]$ ($0 \leq j \leq m-1, 0 \leq t \leq j$), 故 $XY = A_m B_m 2^{2mn/(m+1)} + A_m 2^{mn/(m+1)} S_2 + B_m 2^{mn/(m+1)} S_1 + S_1 S_2$ 通过恒等变形所产生的 $(m+1)(m+2)/2$ 次乘法可表示为 $\left[\sum_{i=t}^j A_i \right] \left[\sum_{i=t}^j B_i \right]$ ($0 \leq j \leq m, 0 \leq t \leq j$), 即 $k=m+1$ 时定理结论成立. 证毕

定理 2 将输入规模为 n 的两个二进制整数 X 和 Y 各分成规模相等的 k ($2 \leq k \leq n$) 部分, 则利用分治算法计算 X 和 Y 的乘积时其计算复杂度为 $O(n^{\lg_k[k(k+1)/2]})$, 从而计算 X 和 Y 的乘积时以 $k=2$ 时的计算复杂度为最小.

证明: 由定理 1 知, 利用分治算法计算 X 和 Y 的乘积时, 可将 k^2 次规模为 n/k 位的二进制整数乘法降为 $k(k+1)/2$ 次, 此外, 加法和移位的次数不超过 cn , 故计算 X 和 Y 乘积的计算复杂度 $T(n)$ 为

$$T(n) = \begin{cases} 1, & n=1 \\ kT(n/k) + cn, & n>1 \end{cases}$$

从而 $T(n) = O(n^{\lg_k[k(k+1)/2]})$.

$k=2$ 时, $T(n) = O(n^{\lg_2 3}) \approx O(n^{1.59})$; $k=3$ 时, $T(n) = O(n^{\lg_3 6}) \approx O(n^{1.63})$; $k=4$ 时, $T(n) = O(n^{\lg_4 10}) \approx O(n^{1.66})$. 显然 $O(n^{\lg_2 3}) < O(n^{\lg_3 6}) < O(n^{\lg_4 10})$. 再由

引理 4, 函数 $f(x) = \log_x[(x+1)/2]$ 在区间 $(3, +\infty)$ 上是严格单调递增的, 从而在 n 给定的情况下, 随着 k 的增大 $n^{\lg_k[k(k+1)/2]}$ 的值也随着增大, 故 $k=2$ 时的计算复杂度最小. 证毕

注: 事实上, 由引理 4 和 $\lim_{k \rightarrow \infty} \log_k[k(k+1)/2] = 2$ 知 $\log_k[k(k+1)/2] < 2$, 显然 $1 < \log_k[k(k+1)/2]$, 故 $1 < \log_k[k(k+1)/2] < 2$, 这说明分治算法与普通乘法相比总是有效的, 但以 $k=2$ 时的计算复杂度为最小.

4 结论

本文通过对大整数相乘问题的研究和分析, 给出了利用分治算法将大整数各分成规模相等的 k ($2 \leq k \leq n$) 部分进行计算时相应的计算复杂度. 研究结果表明, 将输入分成规模相等的两部分时, 其相应的计算复杂度最小, 从而将大整数各分成规模相等的两部分来进行处理是合理的. 本文的结果为实际处理大整数相乘问题时所采用的分治策略提供了一定的理论依据, 对算法的设计与分析也具有重要的意义.

参考文献:

- [1] 朱洪, 陈增武, 段振华, 周克成. 算法设计与分析[M]. 上海: 上海科学技术文献出版社, 1989. 155-156.
Zhu Hong, Chen Zengwu, Duan Zhenhua, Zhou Kecheng. The Design and Analysis of Algorithm[M]. Shanghai: Shanghai Science & Technology Press, 1989. 155-156. (in Chinese)
- [2] Aho A V, Hopcroft J E, Ullman J D. The Design and Analysis of Computer Algorithms[M]. Addison Wesley Publishing Company, 1974.
- [3] Aho A V, Hopcroft J E, Ullman J D. Data Structures and Algorithms[M]. Addison Wesley Publishing Company, 1983.
- [4] 数学分析. 华东师范大学数学系编[M]. 北京: 高等教育出版社, 1991. 47-48.

作者简介:



王念平 男, 1973 年出生于河南, 现为解放军信息工程大学电子技术学院博士研究生. 主要研究方向为密码理论和应用数学.

E-mail: wwnpp@126.com

金晨辉 男, 1965 年出生于河南, 博士, 现为解放军信息工程大学电子技术学院教授, 博士生导师. 主要研究方向为密码学和信息安全. E-mail: jinchenhui@126.com