Multiprecision Integer Division Examples Using Arbitrary Radix

ERIC REGENER
Concordia University, Montreal, Quebec, Canada

In Knuth's algorithm for multiprecision integer division, the estimated quotient digit may differ from the correct value by 2 in rare cases. We derive the entire range of cases in which this situation can occur for a given radix. The results are expected to be useful in testing realizations of the algorithm in computer algebra systems and elsewhere.

Categories and Subject Descriptors: G.1.0 [**Numerical Analysis**]: General—*computer arithmetic*; G.4 [**Mathematics of Computing**]: Mathematical Software—*certification and testing, verification*; I.1.3 [**Algebraic Manipulation**]: Languages and Systems—*special-purpose algebraic systems*

General Terms. Algorithms, Verification

Additional Key Words and Phrases: Integer division, multiprecision arithmetic, computer algebra

1. INTRODUCTION

We present examples that are useful in testing realizations of Knuth's algorithm [1, p. 237] for division of nonnegative integers using an arbitrary radix. We derive the entire range of situations in which the seldom-executed special cases of the algorithm are performed.

We use the notation

$$[a_m, a_{m+1}, \ldots, a_n] = \sum_{j=m}^{n} a_j b^{n-j},$$

where b is the radix. We assume throughout that b is even and $b \geq 6$. We also write $\lfloor x \rfloor$ for the greatest integer less than or equal to x and $\lceil x \rceil$ for the least integer greater than or equal to x.

The algorithm divides $s = [s_{-1}, \ldots, s_m]$ by $v = [v_0, \ldots, v_n]$, for $m, n \geq 1$, $s_{-1} \geq 0$, $s_0 > 0$. It is assumed that $v_0 \geq b/2$; if it is not, it can be made so by multiplying both s and v by $\lfloor b/(v_0 + 1) \rfloor$.

The only tricky part of the algorithm is the estimation of the successive digits q_i of the quotient. For a given quotient digit q, the procedure is as follows, assuming $m = n$:

(1) Estimate q by $\tilde{q} = \lfloor [s_{-1}s_0]/v_0 \rfloor$. (If $\tilde{q} \geq b$, we may adjust it immediately to $b - 1$.)

This work was supported by grants from NSERC of Canada and FCAC of Quebec.
Author's address: Department of Computer Science, Concordia University, 1455 de Maisonneuve Boulevard West, Montreal, Quebec H3G 1M8, Canada.

(2) Adjust $\tilde{q}$ to $\hat{q} = \lfloor [s_{-1}s_0s_1]/[v_0v_1] \rfloor$ by subtracting 1 or 2 if necessary.
(3) Perform the division, replacing s by $s - \hat{q}v$. If this quantity is negative, subtract 1 from $\hat{q}$, producing the true quotient digit q.

Knuth proves that the given conditions $q \le \tilde{q} \le q + 2$ and $q \le \hat{q} \le q + 1$. He also shows that Step 3 will be performed with a probability (in some sense) of $3/b$. If b is as large as the size of a computer word, as it is in the usual applications, this probability is very small. Therefore, it is important to test any realization of the algorithm with cases in which Steps 2 and 3 are both executed. For example, for $b = 10$, $s = 4791$, and $v = 599$, we have $q = \lfloor 4791/599 \rfloor = 7$, while $\hat{q} = \lfloor 479/59 \rfloor = 8$ and $\tilde{q} = \lfloor 47/5 \rfloor = 9$.

We show how to find all cases in which $\tilde{q} = q + 2$ for a given radix b, first without and then with the additional condition that $\hat{q} = q + 1$.

2. RESULTS

We first investigate the conditions under which $\tilde{q} = q + 2$ alone. Looking at the most significant digits only, we assume a three-digit dividend s and a two-digit divisor $v = [v_0v_1]$. If $\tilde{q} = q + 2$, then s must satisfy $s_{\min} \le s \le s_{\max}$, where

$$s_{\min} = v_0b(q + 2), \qquad s_{\max} = v(q + 1) - 1.$$

Elaborating the condition $s_{\max} \ge s_{\min}$ gives

$$v_1(q + 1) \ge v_0b + 1. \tag{*}$$

The maximum value of q, say $q_{\max}$, is $b - 3$ since we want $\tilde{q}$ to be less than b. To find the entire range of v and q which is of interest, we solve the following problems:

A. *Find the range of possible v for a given q.* Setting $v_1 = b - 1$ in (*), the maximum v_0 is

$$\max v_0 = \left\lfloor \frac{(b - 1)(q + 1) - 1}{b} \right\rfloor = \left\lfloor q + 1 - \frac{q + 2}{b} \right\rfloor = q.$$

For a given q and v_0, the minimum v_1 is obviously

$$\min v_1 = \left\lceil \frac{v_0b + 1}{q + 1} \right\rceil.$$

If v_0 is maximum for a given q, then

$$\min v_1 = \left\lceil \frac{bq + 1}{q + 1} \right\rceil = b - 1 = \max v_1.$$

B. *Find all possible v for $q = q_{max}$.* Obviously, $\max v_0 = q_{\max} = b - 3$. For a given v_0,

$$\min v_1 = \left\lceil \frac{v_0b + 1}{b - 2} \right\rceil = v_0 + \left\lceil \frac{2v_0 + 1}{b - 2} \right\rceil = v_0 + 2,$$

since $b/2 \le v_0 \le b - 3$.

C. *Find q_{min}, the minimum of q, and the associated range of v.* From (*), letting $v_0 = b/2$ and $v_1 = b - 1$, we have

$$q_{\min} = \left\lceil \frac{v_0 b + 1}{v_1} - 1 \right\rceil = \left\lceil \frac{b^2 + 2}{2(b-1)} - 1 \right\rceil = \left\lceil \frac{b-1}{2} + \frac{3}{2(b-1)} \right\rceil = \frac{b}{2},$$

since we assume b is greater than or equal to 6 and even. From Part A, if $q = q_{\min}$, then $v_{\max} = v_{\min} = [b/2, b - 1]$.

Now we investigate cases in which $\tilde{q} = q + 2$ and $\hat{q} = q + 1$. We must give s four digits and v three digits, $v = [v_0 v_1 v_2]$. The maximum value of s is, as before,

$$s_{\max} = v(q + 1) - 1.$$

The minimum value of s depends now on two conditions. Define

$$t_0 = v_0 b^2 (q + 2),$$

$$t_1 = (v_0 b + v_1) b (q + 1) = s_{\max} - v_2(q + 1) + 1.$$

In order for $\tilde{q} = q + 2$, we must have $s \geq t_0$, while $\hat{q} = q + 1$ implies $s \geq t_1$. Therefore, in this case, $s_{\min} = \max(t_0, t_1)$. However, since $t_1 < t_0$ is equivalent to

$$v_1(q + 1) < v_0 b,$$

which is inconsistent with (*), we must have $s_{\min} = t_1$ in the cases of interest.

The condition $s_{\max} \geq t_1$ implies only that

$$v_2(q + 1) \geq 1,$$

and therefore, that $v_2 \geq 1$. On the other hand, from $s_{\max} \geq t_0$ we get

$$(v_1 b + v_2)(q + 1) \geq v_0 b^2 + 1. \qquad (**)$$

Solving problems A, B, C again from this inequality, we find no further restriction on the results derived from (*). In fact, no condition is imposed on v_2 beyond $v_2 \geq 1$. So for problem C, if $q = q_{\min} = b/2$, then

$$v_{\max} = [b/2, b - 1, b - 1], \qquad v_{\min} = [b/2, b - 1, 1].$$

We illustrate with examples for $b = 10$ and 1000, labeling the results derived from (*) with B and C, and those from (**) with B′ and C′.

Example 1. $b = 10$.

B. $q = q_{\max} = 7$: $v_{\max} = 79$, $s_{\max} = 631$ and $s_{\min} = 630$.
C. $q = q_{\min} = 5$: $v_{\max} = v_{\min} = 59$, $s_{\max} = 353$ and $s_{\min} = 350$.
B′. $q = q_{\max} = 7$: $v_{\max} = 799$, $s_{\max} = 6391$, and $s_{\min} = 6320$.
C′. $q = q_{\min} = 5$: $v_{\max} = 599$, $s_{\max} = 3593$ and $s_{\min} = 3540$;
$v_{\min} = 591$, $s_{\max} = 3545$ and $s_{\min} = 3540$.

Example 2. $b = 1000$.

B. $q = q_{\max} = 997$: $v_{\max} = 997{,}999$, $s_{\max} = 996{,}003{,}001$ and $s_{\min} = 996{,}003{,}000$.
C. $q = q_{\min} = 500$: $v_{\max} = v_{\min} = 500{,}999$, $s_{\max} = 251{,}000{,}498$ and $s_{\min} = 251{,}000{,}000$.

B′. $q = q_{max} = 997$: $v_{max} = 997{,}999{,}999$, $s_{max} = 996{,}003{,}999{,}001$ and $s_{min} = 996{,}003{,}002{,}000$.

C′. $q = q_{min} = 500$: $v_{max} = 500{,}999{,}999$, $s_{max} = 251{,}000{,}999{,}498$ and $s_{min} = 251{,}000{,}499{,}000$; $v_{min} = 500{,}999{,}001$, $s_{max} = 251{,}000{,}499{,}500$ and $s_{min} = 251{,}000{,}499{,}000$.

The user may determine an appropriate range of v for any q between q_{min} and q_{max}, then verify the quotients for values of s between s_{min} and s_{max}.

REFERENCES

1. KNUTH, D.E. *The Art of Computer Programming.* Vol. 2, *Seminumerical Algorithms.* Addison-Wesley, Reading, Mass., 1969.