About an exercise on modular multiplication

ho boon suan

17 July 2025, updated 18 July 2025

The following exercise was added to the second edition of Volume 2 of *The Art of Computer Programming* in August 1995:

Exercise 3.2.1.1–3. *Many computers do not provide the ability to divide a two-word number by a one-word number; they provide only operations on single-word numbers, such as* $\mathrm{himult}(x,y) = \lfloor xy/w \rfloor$ *and* $\mathrm{lomult}(x,y) = xy \bmod w$*, when* x *and* y *are nonnegative integers less than the word size* w*. Explain how to evaluate* $ax \bmod m$ *in terms of* himult *and* lomult*, assuming that* $0 \le a, x < m < w$ *and that* $m \perp w$*. You may use precomputed constants that depend on* a*,* m*, and* w*.*

Knuth provides the following answer: "Let $a' = aw \bmod m$, and let m' be such that $mm' \equiv 1 \pmod{w}$. Set $y \leftarrow \mathrm{lomult}(a', x)$, $z \leftarrow \mathrm{himult}(a', x)$, $t \leftarrow \mathrm{lomult}(m', y)$, $u \leftarrow \mathrm{himult}(m, t)$. Then we have $mt \equiv a'x \pmod{w}$, hence $a'x - mt = (z-u)w$, hence $ax \equiv z - u \pmod{m}$; it follows that $ax \bmod m = z - u + [z < u]m$."

I will try to give some motivation for his answer (though ultimately like much of math it involves magic that one just gets used to). We want to find $ax \bmod m$, but it is costly to divide by m. The only affordable division operation we have is division by w via the himult operation. As such, we try the multiplication-by-one trick, which gives

$$ax \bmod m = \left(ax \cdot \frac{w}{w}\right) \bmod m = \frac{(aw)x}{w} \bmod m = \frac{a'x}{w} \bmod m,$$

where $a' := aw \bmod m$. (To be precise, we are multiplying by the inverse w^{-1} modulo m, which is justified as $m \perp w$.)

Since $m \perp w$, we have

$$\frac{a'x}{w} \equiv \frac{a'x - mt}{w} \pmod{m}$$

for any integer t. Thus, if we can choose $0 \le t < w$ such that

$$-m \le \frac{a'x - mt}{w} < m \quad \text{and} \quad a'x \equiv mt \pmod{w}, \qquad (*)$$

Here we are actually dividing by w in the rationals rather than multiplying by the inverse w^{-1} modulo m; this is fine precisely since $a'x \equiv mt \pmod{w}$.

we would be done, since we would then have

$$ax \bmod m = \frac{a'x - mt}{w} + [a'x < mt]m = \left\lfloor \frac{a'x}{w} \right\rfloor - \left\lfloor \frac{mt}{w} \right\rfloor + [a'x < mt]m.$$

Now $a'x = wz + y$ where $z \leftarrow \mathrm{himult}(a', x)$ and $y \leftarrow \mathrm{lomult}(a', x)$, so $a'x \equiv y \pmod{w}$. Thus our choice of t must satisfy $mt \equiv y \pmod{w}$, which leads us to set $t \leftarrow \mathrm{lomult}(m', y)$ where m' is such that $mm' \equiv 1 \pmod{w}$. One can then check that $(*)$ holds; setting $u \leftarrow \mathrm{himult}(m, t)$, we conclude that

$$ax \bmod m = z - u + [z < u]m.$$

Remarks. The ideas in this exercise lead to *Montgomery multiplication*, where one works with the *Montgomery forms* $xw \bmod m$ of residue classes $x \bmod m$ instead of working with $x \bmod m$. See Peter L. Montgomery, *Mathematics of Computation* **44** (1985), 519–521, doi:10.1090/S0025-5718-1985-0777282-X. For an overview of modern algorithms for modular multiplication, see Section 2.4 of Richard P. Brent and Paul Zimmermann, *Modern Computer Arithmetic* (Cambridge University Press, 2010). A near-final draft is available online at https://members.loria.fr/PZimmermann/mca/mca-cup-0.5.9.pdf.

Thanks to Way Yan Win for helpful comments on an earlier version of this note.