Integer Multiplication

We would like to multiply two large integers:

$$\begin{array}{r} \mathtt{y_{n-1}y_{n-2}y_{n-3}\cdots y_3y_2y_1y_0} \\ \times\ \underline{\mathtt{x_{n-1}x_{n-2}x_{n-3}\cdots x_3x_2x_1x_0}} \end{array}$$

COMMENT: We will IGNORE carries throughout.

Addition Algorithm

Start with addition:

$$\begin{array}{r} \mathtt{y_{n-1}y_{n-2}y_{n-3}\cdots y_3y_2y_1y_0} \\ +\ \underline{\mathtt{x_{n-1}x_{n-2}x_{n-3}\cdots x_3x_2x_1x_0}} \end{array}$$

The time to add two integers is linear in the number of digits.

Upper bound: Elementary school algorithm.
Lower bound: Any algorithm must examine every digit.

From now on, assume that time to add two n digit numbers is exactly

$$A(n) = \alpha n \text{ (for some constant } \alpha\text{).}$$

“Standard” Multiplication Algorithm

Elementary school algorithm with no additions until after all of the multiplies.

EXAMPLE:

```
    4352
  × 3748
  ------
      16
     40
    24
   32
--------
     08
    20
   12
  16
--------
    14
   35
  21
 28
--------
   06
  15
 09
12
--------
```

Concatenate the “even” and “odd” indexed values within each group then sum by column.

```
    4352
  × 3748
  ------
    2416
   3240
--------
   1208
  1620
--------
  2114
 2835
--------
 0906
1215
--------
16311296
```

Atomic multiplies: n^2.
Every digit on bottom is multiplied with every digit on top.

Atomic additions: $2n(n-1)$.
By column, right-to-left: $0+2+4+6+8+\ldots+2(n-1)+2(n-1)+\ldots+8+6+4+2+0 \;=\; 2\sum_{i=0}^{n-1} 2i$.

Recursive Multiplication Algorithm.

Use divide-and-conquer. Take two n digit numbers x, y and cut each in half to form:

$$x = a \circ b, \qquad y = c \circ d$$

where a is the $n/2$ leftmost digits of x, b is the $n/2$ rightmost digits of x, c is the $n/2$ leftmost digits of y, and d is the $n/2$ rightmost digits of y. Now,

$$xy = ac10^n + (ad + bc)10^{n/2} + bd$$

Multiplying by a power of 10 can be accomplished by shifts (this holds for any base).

EXAMPLE:

```
  4352
× 3748
```

$$a = 37, \quad b = 48, \quad c = 43, \quad d = 52$$

$$ac = 37 \cdot 43 = 1591, \quad ad = 37 \cdot 52 = 1924, \quad bc = 48 \cdot 43 = 2064, \quad bd = 48 \cdot 52 = 2496$$

$$ad + bc = 1924 + 2064 = 3988$$

```
   15912496
+    3988
   16311296
```

So we need the four products ac, ad, bc, and bd, which can be attained by calling the algorithm recursively four times (on $n/2$ digit numbers). The two values ac and bd can be concatenated, and ad can be added to bc in time αn. The final result is the sum of $ac \circ bd$ and $ad + bc$. Ignoring the rightmost $n/2$ digits of $ac \circ bd$, this sum can digit attained with and additional αn time plus the cost of potentially $n/2$ carries (which for simplicity we ignore). So the total time for the additions is $\alpha 2n$. The recursion ends when $n = 1$ and it multiplies two one digit numbers.

The recurrence for the time to multiply:

$$M(n) = 4M\left(\frac{n}{2}\right) + 2\alpha n$$

and $M(1) = \mu$. Using the tree method we find that

$$M(n) = \mu n^2 + 2\alpha n(n-1)$$

This is still quadratic and pretty much matches the elemtary school algorithm.

NOTE: In "real life" you would not recurse down to one digit, but down to the word size where the computer can do an atomic multiplication. (This comment holds for additions as well.) You can think of this as doing arithmetic in base 2^w, where w is the word size.

Multiplying two two-digit numbers.

Standard Algorithm. EXAMPLE:

```
   52
×  36
   12
  30
  06
 15
 1872
```

Four atomic multiplications and four atomic additions.

Clever Algorithm.

Two two-digit numbers can be multiplied using only *three* atomic multiplications!!!

```
   cd
×  ab
```

Form

$$\begin{aligned} w &= (a+b)(c+d) \\ u &= ac \\ v &= bd \end{aligned}$$

Note that

$$w = (a+b)(c+d) = ac + ad + bc + bd = ac + (ad+bc) + bd$$

So

$$w - (u+v) = [ac + (ad+bc) + bd] - [ac+bd] = ad + bc$$

Just what we want!!!

The full product is

$$xy = u10^2 + (w - (u+v))10 + v$$

EXAMPLE:

$$\begin{aligned} w &= (a+b)(c+d) = (3+6)(5+2) = 9 \cdot 7 = 63 \\ u &= ac = 3 \cdot 5 = 15 \\ v &= bd = 6 \cdot 2 = 12 \end{aligned}$$

$$xy = u10^2 + (w-(u+v))10 + v = 15 \cdot 100 + (63 - (15+12))10 + 12 = 15 \cdot 100 + 36 \cdot 10 + 12 = 1872$$

```
   52
×  36
   12
  36
 15
```

We have reduced the number of atomic multiplies from four to three, at the cost of increasing the number of atomic additions from four to eight.

Putting it all together

Recall

$$\begin{aligned} w &= (a+b)(c+d) \\ u &= ac \\ v &= bd \end{aligned}$$

Then the product is

$$xy = u10^n + (w - (u+v))10^{n/2} + v$$

EXAMPLE for four digit numbers.

$$\begin{array}{r} 4352 \\ \times\ \underline{3748} \end{array}$$

$$\begin{aligned} w &= (a+b)(c+d) = (37+48)(43+52) = 85 \cdot 95 = 8075 \\ u &= ac = 37 \cdot 43 = 1591, \\ v &= bd = 48 \cdot 52 = 2496, \\ u+v &= ac+bd = 1591 + 2496 = 4087 \\ w-(u+v)) &= ad+bc = 8075 - 4087 = 3988 \end{aligned}$$

Now it is the same as the previous example.

We can estimate the addition time as $\alpha n/2$ to form $a+b$, $\alpha n/2$ to form $c+d$, αn to form $v+w$, αn to subtract that sum from u, and αn to add that to $v \circ w$. The total is $4\alpha n$. The recurrence for the time to multiply:

$$M(n) = 3M\left(\frac{n}{2}\right) + 4\alpha n$$

and $M(1) = \mu$. Using the tree method we find that

$$M(n) = (\mu + 8\alpha)n^{\lg 3} - 8\alpha n$$