We would like to multiply two large integers:

\[ y_{n-1}y_{n-2}y_{n-3} \cdots y_3y_2y_1y_0 \]

\[ \times \quad \underline{x_{n-1}x_{n-2}x_{n-3} \cdots x_3x_2x_1x_0} \]

We will ignore carries throughout.
“Standard” Multiplication Algorithm

\[
\begin{array}{c}
4352 \\
\times 3748 \\
\hline
16 \\
40 \\
24 \\
32 \\
\hline
08 \\
20 \\
12 \\
16 \\
\hline
14 \\
35 \\
21 \\
28 \\
\hline
06 \\
15 \\
09 \\
12 \\
\hline
\end{array}
\]

Concatenate the ‘‘even’’ and ‘‘odd’’ indexed values.

\[
\begin{array}{c}
4352 \\
\times 3748 \\
\hline
2416 \\
3240 \\
\hline
2114 \\
2835 \\
\hline
0906 \\
1215 \\
\hline
16311296 \\
\end{array}
\]

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:
\[
0 + 2 + 4 + 6 + 8 + \ldots + 2(n - 1) + 2(n - 1) + \ldots + 8 + 6 + 4 + 2 + 0 = 2 \sum_{i=0}^{n-1} 2i.
\]
Two-digit multiplication

\[
\begin{array}{c}
\text{cd} \\
\times \text{ab}
\end{array}
\]

Example

\[
\begin{array}{c}
73 \\
\times 48
\end{array}
\]

\[
a = 4, \quad b = 8, \quad c = 7, \quad d = 3
\]

\[
ac = 4 \cdot 7 = 28, \quad ad = 4 \cdot 3 = 12, \quad bc = 8 \cdot 7 = 56, \quad bd = 8 \cdot 3 = 24
\]

\[
ad + bc = 12 + 56 = 68
\]

\[
2824 \\
+ 68
\]

\[
3504
\]
“Every” multiplication is a two-digit multiplication ... with a large enough base.

Example

Base 100

\[
\begin{array}{c}
4352 \\
\times \ 3748 \\
\end{array}
\]

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

\[ac = 37 \cdot 43 = 1591, \quad ad = 37 \cdot 52 = 1924,\]

\[bc = 48 \cdot 43 = 2064, \quad bd = 48 \cdot 52 = 2496\]

\[ad + bc = 1924 + 2064 = 3988\]

\[
\begin{array}{c}
15912496 \\
+ \ 3988 \\
\hline
16311296 \\
\end{array}
\]
Another two-digit multiplication.

Example (Base 1000)

\[
\begin{array}{c}
439052 \\
\times \quad 371448 \\
\end{array}
\]

\[a = 371, \quad b = 448, \quad c = 439, \quad d = 052\]
Recursive Multiplication Algorithm

\[
\begin{array}{c}
\underbrace{y_{n-1} \cdots y_{n/2}}_c \underbrace{y_{n/2-1} \cdots y_0}_d \\
\times \underbrace{x_{n-1} \cdots x_{n/2}}_a \underbrace{x_{n/2-1} \cdots x_0}_b
\end{array}
\]

\[
x = a \circ b, \quad y = c \circ d
\]

Product:

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

Recurrence for the time to multiply:

\[
M(n) = 4M\left(\frac{n}{2}\right) + 4A\left(\frac{n}{2}\right)
\]

\[
= 4M\left(\frac{n}{2}\right) + 2\alpha n, \quad M(1) = \mu.
\]

Solve with tree method:

\textbf{Do in class.}
In our examples, an atomic value is a base 10 number.

What is an atomic value for “real life” computer?
Can we do better?

Why or why not?
Can we do better?

Why or why not?

Andrey Kolmogorov
Can we do better?

Why or why not?

Andrey Kolmogorov
Anatoly Karatsuba
Returning to two-digit multiplication.

Standard Algorithm.

Example

\[
\begin{array}{c}
52 \\
\times \\
36 \\
\end{array}
\]

\[
\begin{array}{c}
\underline{12} \\
30 \\
06 \\
\end{array}
\]

\[
\begin{array}{c}
15 \\
\end{array}
\]

\[
\begin{array}{c}
1872 \\
\end{array}
\]

Four atomic multiplications and four atomic adds.
Clever Algorithm

Only need *three* atomic multiplications!!

\[
\begin{align*}
\text{cd} \\
\times \quad \text{ab}
\end{align*}
\]

Form

\[
ac, \quad bd, \quad w = (a + b)(c + d)
\]

Note that

\[
w = (a + b)(c + d) = ac + ad + bc + bd = ac + (ad + bc) + bd
\]

So

\[
w - (ac + bd) = [ac + (ad + bc) + bd] - [ac + bd] = ad + bc
\]

Just what we want!!!

The full product is

\[
xy = ac10^2 + (w - (ac + bd))10 + bd
\]
Example

\[ w = (a + b)(c + d) = (3 + 6)(5 + 2) = 9 \cdot 7 = 63 \]
\[ ac = 3 \cdot 5 = 15, \quad bd = 6 \cdot 2 = 12 \]

\[ xy = ac10^2 + (w - (ac + bd))10 + bd \]
\[ = 15 \cdot 100 + (63 - (15 + 12))10 + 12 \]
\[ = 15 \cdot 100 + 36 \cdot 10 + 12 = 1872 \]

\[
\begin{array}{c}
52 \\
\times \ \ 36 \\
\hline
12 \\
\hline
36 \\
\hline
15 \\
\hline
1872
\end{array}
\]

\[
\begin{array}{c}
52 \\
\times \ \ 36 \\
\hline
1512 \\
\hline
36 \\
\hline
1872
\end{array}
\]
Putting it all together
Recall that we have to form

\[ ac, \quad bd, \quad w = (a + b)(c + d) \]

Then the product is

\[ xy = ab10^n + (w - (ac + bd))10^{n/2} + bd \]

Three atomic multiplies and eight atomic adds.

Recurrence for the time to multiply:

\[
M(n) = 3M\left(\frac{n}{2}\right) + 8A\left(\frac{n}{2}\right) \\
= 3M\left(\frac{n}{2}\right) + 4\alpha n \quad M(1) = \mu.
\]

Solve with tree method: Do in class.