Lec 1 2023 Integer Multiplication

January 4, 2023 1:39 PM

Integer Multiplication

Let a and b be two long integers.
How do we multiply axb on a computer?
How fast can we multiply?

64 bit computer
a b
64bit x 64bit = 128bit.
$-2^{63} \le a < 2^{63}$
$0 \le a < 2^{64}$

$$\begin{array}{r} 345 = a \\ \times \quad 73 = b \\ \hline 1035 \quad = 3 \times 345 \\ 24150 \quad = 70 \times 345 \\ \hline 25185 \end{array}$$

Suppose a has m digits and b has n digits. We did m×n single digit multiplications and < 2mn additions.

Use arrays. $a =$ [5|4|3] (indices 0 1 2)

$b =$ [3|7]

m=3 n=b

$a \cdot b \le 999 \cdot 99 < 10^5$

$ab < 10^{m+n}$

$C =$ [0|0|0|0|0]

3x345 $C =$ [5|3|0|1|]

70 x 345 $C =$ [5|8|1|5|2]

$t = 7 \cdot 5 + 3 = 38$ (carry)

$t = 7 \cdot 4 + 0 + 3 = 31$ (carry)

$t = 7 \cdot 3 + 1 + 3 = 25$ (carry.)

```
> IntMul := proc( m::posint, a::Array,
                  n::posint, b::Array,
                  c::Array, # space for product
                  B::posint )
      local i,j,t,carry;
      for i from 0 to m+n-1 do
         c[i] := 0;
      od;
      for i from 0 to m-1 do
         carry := 0;
         for j from 0 to n-1 do
            t := b[i]*a[j] + c[i+j] + carry;
            c[i+j] := irem(t,B,'carry');
         od;
         c[i+j] := carry;
      od;
      # return the length of the product
      if carry=0 then n+m-1 else n+m fi;
   end:
```

returns the # digits in C.

$\leq k_4(m+n)$.

b

a

$\left.\begin{array}{l} t := b[i]*a[j] + c[i+j] + carry; \\ c[i+j] := irem(t,B,'carry'); \end{array}\right] \leq k_1 m \cdot n$

remainder $\div B$; quotient

$\leq k_2 m$

$\left.\begin{array}{l} \text{if carry=0 then n+m-1 else n+m fi;} \end{array}\right] \leq k_3$

How long does this code take a function of m and n?

```
> a := Array(0..2,[5,4,3]): # a = 345
  b := Array(0..2,[6,7,8]): # b = 876
> c := Array(0..5):
  n := IntMul(3,a,3,b,c,10);
```

$n := 6$

```
> 345*876 = add( c[i]*10^i, i=0..n-1 );
```

$302220 = 302220$

$$T(m,n) \leq k_1 mn + k_2 m + k_3 + k_4(m+n)$$
$$= k_1 mn + k_5 m + k_4 n + k_3 \quad \text{where } k_5 = k_2 + k_4$$
$$\sim k_1 mn \quad \text{(for large } m \text{ and } n\text{)}.$$
$$T(m,n) \in O(mn)$$

This means if we double the #digits of a and b

i.e. $n \to 2n$, $m \to 2m$ then $\dfrac{T(2m,2n)}{T(m,n)} = \dfrac{k_1(2m\,2n) + k_5(2m) + k_4(2n) + k_3}{k_1 mn + k_5 m + k_4 n + k_3}$

$$\lim_{\substack{n\to\infty \\ m\to\infty}} \frac{T(2m,2n)}{T(m,n)} = \frac{k_1 4mn + \cdots}{k_1 mn + \cdots} = 4.$$

So for large m & n the cost (time) goes up by a factor of 4.

If $n=m$ then $T(n) = k_1 n^2 + (k_5+k_4)n + k_3 \in O(n^2)$

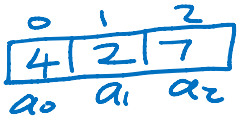
i.e. $T(n)$ is quadratic.

i.e. $T(n)$ is quadratic.

Let $B > 1$ be the base of our integers.

Let $a = \sum_{i=0}^{n-1} a_i B^i$ where $0 \le a_i < B$. Then $0 \le a < B^n$.

We use $B = 10$. $724 = \underset{a_0}{4} + \underset{a_1}{2} \cdot 10 + \underset{a_2}{7} \cdot 1$



GMP = Gnu MultiPrecision library.

$B = 2^{64}$. $a =$ 

64bits 64bits 64bits.

How big can t be? I Claim $t < B^2$ $B = 10$ $B < 100$.

$t = \underset{<B}{a[i]} * \underset{<B}{b[j]} + \underset{<B}{c[i+j]} + \underset{<B}{carry}$?

$\le (B-1)^2 + (B-1) + (B-1)$

$= B^2 - 2B + 1 + 2B - 2 = B^2 - 1 < B^2$

$carry = t/B < B$