3 Karatsuba’s Multiplication

x : n-digit no. $\quad$ y : n-digit no. $\quad$ x_M : First-half $\quad$ x_L : 2nd half

$$x \cdot y = (x_M \cdot 10^{n/2} + x_L) \cdot (y_M \cdot 10^{n/2} + y_L)$$

$$x \cdot y = x_M \cdot y_M \cdot 10^n + x_M \cdot y_L \cdot 10^{n/2} + x_L \cdot y_M \cdot 10^{n/2} + x_L \cdot y_L$$

$$x \cdot y = x_M \cdot y_M \cdot 10^n + (\underbrace{x_M \cdot y_L + x_L \cdot y_M}_{z}) \cdot 10^{n/2} + x_L \cdot y_L$$

$$(x_M + x_L) \cdot (y_M + y_L) = x_M \cdot y_M + x_M \cdot y_L + x_L \cdot y_M + x_L \cdot y_L$$

$$\underbrace{x_M \cdot y_L + x_L \cdot y_M}_{z} = \underbrace{((x_M + x_L) \cdot (y_M + y_L))}_{c} - (\underbrace{x_M \cdot y_M}_{a} + \underbrace{x_L \cdot y_L}_{b})$$

$a = x_M \cdot y_M$ — 1

$b = x_L \cdot y_L$ — 2

$c = (x_M + x_L) \cdot (y_M + y_L)$ — 3

Only 3 recursive calls needed!

$$x \cdot y = a \cdot 10^n + z \cdot 10^{n/2} + b$$

Ex)

$x = 123 \qquad y = 345 \qquad n = 3$

$x : 123 \qquad x_m = 12 \qquad x_l = 3$

$y : 345 \qquad y_m = 34 \qquad y_l = 5$

$n = 2$ because odd

$$b = x_l \cdot y_l = 3 \cdot 5 = 15$$

$$a = x \cdot y = 12 \cdot 34 = 408$$

$x' \qquad y'$

$x'_m = 1 \qquad x'_l = 2$

$y'_m = 3 \qquad y'_l = 4$

$n = 2$ because even

$a' = x'_m \cdot y'_m = 3$

$b' = x'_l \cdot y'_l = 8$

$c' = (x'_m + x'_l) \cdot (y'_m + y'_l) = (1+2) \cdot (3+4) =$

$z' = c' - (a' + b') = 21 - (3+8) = 10$

$res = a' \cdot 10^{n'} + z \cdot 10^{n/2} + b$

$res = 3 \cdot 10^2 + 10 \cdot 10^1 + 8 = 408$

$$c = (x_m + x_l) \cdot (y_m + y_l) = (12 + 3) \cdot (34 + 5)$$
$$= 15 \cdot 39 = 585$$

$\overset{\downarrow}{x'}, \quad \overset{\downarrow}{y'}$

$res = 583$

$$z = c - (a+b)$$
$$= 585 - (15 + 408) = 162$$

$$res = a \cdot 10^n + z \cdot 10^{n/2} + b$$
$$= 408 \cdot 10^2 + 162 \cdot 10^1 + 15 = 42435$$

```
def Kar(x, y):
    if n=1:
        return x·y          } —— b
    else:
        xM, yM = first half of x, y
        xL, yL = second half of x, y

        a = Kar(xM, yM)  —— T(n/2)
        b = Kar(xL, yL)  —— T(n/2)     3T(n/2)+c
```

$c = Kar((x_m + x_l), (y_m + y_l))$ — $T(n/2)$

$z = c - (a+b)$

$res = a \cdot 10^n + z \cdot 10^{n/2} + b$ — $c \cdot n$

return res

This is the algorithm used in Python to multiply large numbers. Don't underestimate the difference between n^2 (blue line in Figure 2.2) and $n^{1.59}$ (green line)!

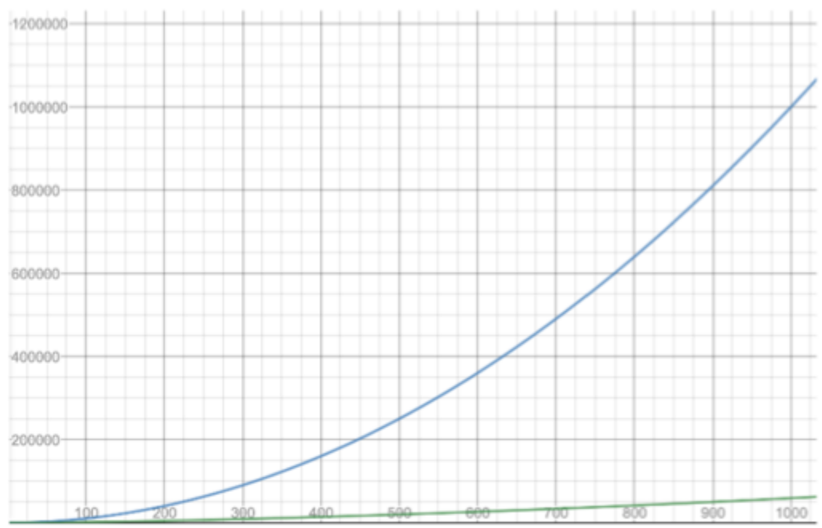

Figure 2.2: Growth of n^2 and $n^{\log_2 3}$.

$T(n) \leq 3T(n/2) + c \cdot n$

$T(n) = O(n^{\log_2 3}) = O(n^{1.59})$

Time: $O(n^{\log_2 3})$

Space: $O(n)$

Code:

```
def karatsuba(x, y):

  if x < 10 and y < 10:
    return x * y
```

```
n = max(len(str(x)), len(str(y)))
if ( n%2 != 0): n-=1
half_n = n//2

x_m, x_l = x // (10**half_n), x % (10**half_n)
y_m, y_l = y // (10**half_n), y % (10**half_n)

a = karatsuba( x_m, y_m )
b = karatsuba( x_l, y_l )
c = karatsuba( x_m+x_l, y_m+y_l )
z = c-(a+b)

res = (a*10**n) + (z*10**half_n) + b
return res
```