Complexity of algebraic computation
Some divide and conquer algorithms
Proposition  Let $a$, $b$, $c$, and $d$ be positive integers. The solution to the recurrence

$$T(n) = \begin{cases} 
  b, & \text{for } n = 1 \\
  aT(n/c) + bn^d, & \text{for } n > 1
\end{cases}$$

(for $n$ being a power of $c$) is

$$T(n) = \begin{cases} 
  O(n^d), & \text{if } a < c^d \\
  O(n^d \log n), & \text{if } a = c^d \\
  O(n^{\log_c a}), & \text{if } a > c^d
\end{cases}$$

Lemma

$$T(c^k) = bc^{kd} \sum_{i=0}^{k} \left( \frac{a}{cd} \right)^i$$
Proof The proof is by induction on $k$. The basis ($k = 0$) is trivial (how?) and for the induction step assume that the equality holds for $k$. Then

$$T(c^{k+1}) = aT(c^k) + bc^{(k+1)d}$$

$$= abc^{kd} \sum_{i=0}^{k} \left( \frac{a}{cd} \right)^i + bc^{(k+1)d}$$

$$= bc^{(k+1)d} \left( \sum_{i=0}^{k} \left( \frac{a}{cd} \right)^i + 1 \right)$$

$$= bc^{(k+1)d} \left( \sum_{i=1}^{k+1} \left( \frac{a}{cd} \right)^i + 1 \right)$$

$$= bc^{(k+1)d} \sum_{i=0}^{k+1} \left( \frac{a}{cd} \right)^i.$$

$\square$
Corollary

\[ T(n) = bn^d \sum_{i=0}^{k} \left( \frac{a}{c^d} \right)^i \]

**Proof of the proposition** If \( a = c^d \), then

\[ T(n) = bn^d \sum_{i=0}^{k} \left( \frac{a}{c^d} \right)^i = bn^d \sum_{i=0}^{k} 1 = O(n^d \log n). \]

Otherwise,

\[ T(n) = bn^d \sum_{i=0}^{k} \left( \frac{a}{c^d} \right)^i = bn^d \left( \frac{a/c^d}{(a/c^d) - 1} \right)^{\log_e n + 1} - 1, \]

which is \( O(n^d) \), if \( a < c^d \) (why?).
Finally, if \( a > c^d \), then

\[
T(n) = bn^d \frac{(a/c^d)^{\log_c n+1} - 1}{(a/c^d) - 1} < \frac{bn^d}{(a/c^d) - 1} (a/c^d)^{\log_c n+1}
\]

\[
= \frac{a}{c^d} \frac{bn^d}{(a/c^d) - 1} n^{\log_c a} = \frac{ab}{a - c^d} n^{\log_c a}
\]

\[
= O(n^{\log_c a}) .
\]
Multiplication of polynomials
\[
x(\lambda) = \sum_{i=0}^{n-1} x_i \lambda^i,
\]

\[
y(\lambda) = \sum_{j=0}^{n-1} y_j \lambda^j, \text{ and}
\]

\[
z(\lambda) = x(\lambda)y(\lambda) = \sum_{k=0}^{2n-2} z_k \lambda^k.
\]

That is \(z_k = \sum_{i+j=k} x_i y_j\), \(k = 0, 1, \ldots, 2n - 2\).
The computation task

Given $x_0, x_1, \ldots, x_{n-1}$ and $y_0, y_1, \ldots, y_{n-1}$, compute

\begin{align*}
    z_0 &= x_0 y_0 \\
    z_1 &= x_0 y_1 + x_1 y_0 \\
    &\vdots & \vdots & \vdots \\
    z_{n-1} &= x_0 y_{n-1} + x_1 y_{n-2} + \cdots + x_{n-1} y_0 \\
    z_n &= x_1 y_{n-1} + \cdots + x_{n-1} y_1 \\
    &\vdots & \vdots \\
    z_{2n-2} &= x_{n-1} y_{n-1} \\
\end{align*}

A straightforward computation requires $n^2$ multiplications and $(n - 1)^2$ additions.
Example

\[(x_1 \lambda + x_0)(y_1 \lambda + y_0)\]
\[= x_1 y_1 \lambda^2 + (x_0 y_1 + x_1 y_0) \lambda + x_0 y_0\]
\[= x_1 y_1 \lambda^2 + ((x_0 + x_1)(y_0 + y_1) - x_1 y_1 - x_0 y_0) \lambda + x_0 y_0\]

This computation is performed in 3 multiplications (which ones?) and 4 additions/subtractions.

Let \(M(n)\) denote the number of algebraic operations needed to compute the product of two polynomials of degree \(n - 1\).
Assume that \( n \) is even and let
\[
x_0(\lambda) = \sum_{i=0}^{\frac{n}{2}-1} x_i \lambda^i, \quad x_1(\lambda) = \sum_{i=0}^{\frac{n}{2}-1} x_{i+\frac{n}{2}} \lambda^i,
\]
\[
y_0(\lambda) = \sum_{j=0}^{\frac{n}{2}-1} y_j \lambda^j, \quad y_1(\lambda) = \sum_{j=0}^{\frac{n}{2}-1} y_{j+\frac{n}{2}} \lambda^j.
\]
Then
\[
x(\lambda)y(\lambda)
= (x_1(\lambda)\lambda^{\frac{n}{2}} + x_0(\lambda))(y_1(\lambda)\lambda^{\frac{n}{2}} + y_0(\lambda))
= x_1(\lambda)y_1(\lambda)\lambda^n
+ ((x_0(\lambda) + x_1(\lambda))(y_0(\lambda) + y_1(\lambda)) - x_1(\lambda)y_1(\lambda) - x_0(\lambda)y_0(\lambda))\lambda^{\frac{n}{2}}
+ x_0(\lambda)y_0(\lambda)
\]
Thus, \( M(n) \leq 3M(\frac{n}{2}) + O(n) \), implying \( M(n) = O(n^{\log_2 3}) = O(n^{1.59}) \).
Multiplication of integers
Let $x = \sum_{i=0}^{n-1} x_i 2^i$ and $y = \sum_{i=0}^{n-1} y_j 2^j$.

A “naive” computation of $xy = \sum_{j=0}^{n-1} y_j x 2^j$ requires $O(n^2)$ bit operations.
Assume that $n$ is even and let

$$x = x_12^{\frac{n}{2}} + x_0 \quad y = y_12^{\frac{n}{2}} + y_0$$

Then

$$xy = x_1y_12^n + (x_0y_1 + x_1y_0)2^{\frac{n}{2}} + x_0y_0$$

The products $x_0y_0$, $x_1y_1$, and the sum of products

$$x_0y_1 + x_1y_0 = (x_1 + x_0)(y_1 + y_0) - x_0y_0 - x_1y_1$$

can be computed in three multiplications of integers of length $\frac{n}{2}$ and $O(n)$ additional bit operations (for additions and subtractions).

Let $M(n)$ denote the number of bit operations need for computing the product of two $n$-bit integers.\(^1\) Then $M(n) = O(n\log_2 3) = O(n^{1.59})$.

\(^1\)The number of operations need for computing the product of two objects will be always denoted by $M$. It is clear from the context which objects are dealt with.
Back to multiplication of polynomials: an optimal algorithm
Alternatively, the coefficients of the product of two polynomials of degree \( n - 1 \) can be computed as follows.

1. Fix any 2\(n-1\) pairwise distinct field elements (points) \( t_0, t_1, \ldots, t_{2n-2} \) and evaluate \( x(\lambda) \) and \( y(\lambda) \) at these points.

2. Compute \( z(t_k) = x(t_k)y(t_k), \ k = 1, 2, \ldots, 2n - 1 \).

3. Interpolate the coefficients of \( z(\lambda) \).

This algorithm requires only 2\(n - 1\) multiplications.
Multiplication of matrices
Let
\[ X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \quad Y = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \]
and let
\[ XY = Z = \begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix} = \begin{pmatrix} x_{11}y_{11} + x_{12}y_{21} & x_{11}y_{12} + x_{12}y_{22} \\ x_{21}y_{11} + x_{22}y_{21} & x_{21}y_{12} + x_{22}y_{22} \end{pmatrix}. \]

**Computation task:** Given \( \{x_{11}, x_{12}, x_{21}, x_{22}\} \) and \( \{y_{11}, y_{12}, y_{21}, y_{22}\} \), compute

\[
\begin{align*}
    z_{11} &= x_{11}y_{11} + x_{12}y_{21} \\
    z_{12} &= x_{11}y_{12} + x_{12}y_{22} \\
    z_{21} &= x_{21}y_{11} + x_{22}y_{21} \\
    z_{22} &= x_{21}y_{12} + x_{22}y_{22}
\end{align*}
\]

A “naive” computation requires 8 multiplications and 4 additions.
\[\begin{align*}
m_1 &= (x_{12} - x_{22})(y_{21} + y_{22}) \\
m_2 &= (x_{11} + x_{22})(y_{11} + y_{22}) \\
m_3 &= (x_{11} - x_{21})(y_{11} + y_{12}) \\
m_4 &= (x_{11} + x_{12})y_{22} \\
m_5 &= x_{11}(y_{12} - y_{22}) \\
m_6 &= x_{22}(y_{21} - y_{11}) \\
m_7 &= (x_{21} + x_{22})y_{11} \\
\end{align*}\]

\[\begin{align*}
z_{11} &= m_1 + m_2 - m_4 + m_6 \\
z_{12} &= m_4 + m_5 \\
z_{21} &= m_6 + m_7 \\
z_{22} &= m_2 - m_3 + m_5 - m_7 \\
\end{align*}\]

Here we have 7 multiplications and 18 additions/subtractions.
Let $X = (x_{ik})$ and $Y = (y_{kj})$ be $n \times n$ matrices:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ \vdots & \vdots & & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{in} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{pmatrix} \quad Y = \begin{pmatrix} y_{11} & \cdots & y_{1j} & \cdots & y_{1n} \\ y_{21} & \cdots & y_{2j} & \cdots & y_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ y_{n1} & \cdots & y_{nj} & \cdots & y_{nn} \end{pmatrix}$$

and let $XY = (z_{ij})$, where $z_{ij} = \sum_{k=1}^{n} x_{ik} y_{kj}$. 
Assume that \( n \) is even and partition the matrices \( X, Y, \) and \( Z = XY \) as follows.

\[
X = \begin{bmatrix}
X_{11} & X_{12} \\
X_{21} & X_{22}
\end{bmatrix}
\quad Y = \begin{bmatrix}
Y_{11} & Y_{12} \\
Y_{21} & Y_{22}
\end{bmatrix}
\quad Z = \begin{bmatrix}
Z_{11} & Z_{12} \\
Z_{21} & Z_{22}
\end{bmatrix},
\]

where \( X_{ij}, Y_{ij}, \) and \( Z_{ij} \) are \( \frac{n}{2} \times \frac{n}{2} \) matrices, \( i, j = 1, 2. \)

Then

\[
\begin{align*}
Z_{11} &= X_{11}Y_{11} + X_{12}Y_{21} \\
Z_{12} &= X_{11}Y_{12} + X_{12}Y_{22} \\
Z_{21} &= X_{21}Y_{11} + X_{22}Y_{21} \\
Z_{22} &= X_{21}Y_{12} + X_{22}Y_{22}.
\end{align*}
\]
\[
\begin{array}{ccc}
X_{11} & X_{12} & i \\
\hline
\times & Y_{12} & j \\
\hline
Y_{22} & & \\
\hline
\end{array}
\]

= 

\[
\begin{array}{ccc}
\bullet & \text{z}_{ij} & \\
\hline
\end{array}
\]
Now $M(n)$ denotes the minimal number of operations (additions and multiplications) needed for computing the product of two $n \times n$ matrices. Then

$$M(n) \leq 7M(n/2) + 18(n/2)^2,$$

implying $M(n) = O(n \log_2 7) = O(n^{2.81}).$

**Note** If $n$ is not a power of 2, then we apply the above procedure to the $2^{\lceil \log_2 n \rceil} \times 2^{\lceil \log_2 n \rceil}$ matrices depicted below. The order of magnitude remains the same.

\[
\begin{array}{ccc}
X & \times & Y \\
0 & & 0 \\
\end{array}
= 
\begin{array}{ccc}
0 \\
\end{array}
\]