The Schönhage-Strassen algorithm for integer multiplication
Let $u$ and $v$ be non-negative integers in the binary representation of length $n$. We shall compute $\text{rem}(uv, 2^n + 1)$ in $O(n \log n \log \log n)$ bit operations.

**Note** From $\text{rem}(uv, 2^{2n} + 1)$ we can compute the product $uv$ itself in $O(2n \log 2n \log \log 2n) = O(n \log n \log \log n)$ bit operations, when we multiply $\underbrace{0 \cdots 0}_n u$ and $\underbrace{0 \cdots 0}_n v$ modulo $2^{2n}$.

**Note** We assume that $n$ is a power of 2; $n = 2^m$ (why may we assume that?).

Let $b = 2^{\lfloor m/2 \rfloor}$ and let $\ell = n/b$. Then $\ell \geq b$ and $b$ divides $\ell$: $\ell/b$ equals 1 or 2.
Let 
\[ u = \sum_{i=0}^{b-1} u_i 2^\ell i \quad \text{and} \quad v = \sum_{j=0}^{b-1} v_j 2^\ell j, \]
where \( u_i \)'s and \( v_j \)'s are \( \ell \)-bit integers.

\[ u: \quad u_b-1 \quad \ldots \quad u_1 \quad u_0 \quad \quad \quad \quad \quad \quad v: \quad v_b-1 \quad \ldots \quad v_1 \quad v_0 \]

Then \( uv = \sum_{k=0}^{2b-2} y_k 2^\ell k \), where \( y_k = \sum_{i+j=k} u_i v_j, k = 0, 1, \ldots, 2b - 2 \).

Since \( n = b\ell, 2^{b\ell} \equiv -1 \mod 2^n + 1 \), and
\[ uv \equiv \sum_{k=0}^{b-1} w_k 2^\ell k \mod 2^n + 1, \]
where \( w_k = y_k - y_{k+b}, k = 0, 1, \ldots, b - 1 \), and \( y_{2b-1} = 0 \).

Since \( 0 \leq y_k < b2^{2\ell} \) (why?), \( |w_k| < b2^{2\ell} \), \( k = 0, 1, \ldots, b - 1 \).
Therefore, after computing the $w_k$’s, $k = 0, 1, \ldots, b - 1$, we need only $O(n)$ bit additions to compute $\text{rem}(uv, 2^n + 1)$:

This is because the length of $w_k$ does not exceed $2\ell + \log b$, and, therefore

$$b(2\ell + \log b) = 2b\ell + 2b \log b = O(n).$$
Computation of $w_k$
Let
\[ f(\lambda) = \sum_{i=0}^{b-1} u_i \lambda^i \quad \text{and} \quad g(\lambda) = \sum_{j=0}^{b-1} v_j \lambda^j. \]

Then the coefficients of
\[ h(\lambda) = \text{rem}(f(\lambda)g(\lambda), \lambda^b + 1) \]
are the \( w_k \)'s:

if \( f(\lambda)g(\lambda) = \sum_{k=0}^{2b-2} y_k \lambda^k \), where \( y_k = \sum_{i+j=k} u_i v_j \), then, since
\[ \lambda^b = -1 \mod \lambda^b + 1, \]
we have
\[ f(\lambda)g(\lambda) \equiv_{\lambda^b+1} \sum_{k=0}^{b-1} (y_k - y_{k+b}) \lambda^k = \sum_{k=0}^{b-1} w_k \lambda^k = h(\lambda). \]
\[ w_k = y_k - y_{k+b} = \sum_{i+j=k} u_i v_j - \sum_{i+j=k+b} u_i v_j \]

It follows that
\[ ((k+1) - b)2^\ell < w_k < (k+1)2^\ell \]
(how?).

In particular, as we have already seen (where?),
\[ |w_k| < b2^\ell. \]

We compute \( w_k \) modulo \( b(2^\ell + 1) \) and then we insert the residue into the appropriate interval: \([1, (k+1)2^\ell]\) or \(((k+1) - b)2^\ell, 0]\) by shifting by a multiple of \( b2^\ell \).

Since \((b, 2^\ell + 1) = 1\), we first compute
\[ w'_k = \text{rem}(w_k, b) \quad \text{and} \quad w''_k = \text{rem}(w_k, 2^\ell + 1), \]
and then compute \( w_k \) from \( w'_k \) and \( w''_k \) using the Chinese remainder theorem.
Computation of $w_k$ from $w'_k$ and $w''_k$
Proposition \[ w_k = (2^{2\ell} + 1) \text{rem}(w'_k - w''_k, b) + w''_k. \]

Proof By the Chinese remainder theorem, it suffices to show that

- the right-hand side of the equality is equivalent to \( w'_k \) modulo \( b \) and
- is equivalent to \( w''_k \) modulo \( 2^{2\ell} + 1 \)

(why?).

The equivalence

\[
(2^{2\ell} + 1) \text{rem}(w'_k - w''_k, b) + w''_k \equiv w'_k \pmod{2^{2\ell} + 1}
\]

is immediate (why?) and, since \( b \) is a power of 2 and \( b \leq \ell \), it divides \( 2^{2\ell} \), implying

\[
2^{2\ell} + 1 \equiv 1 \pmod{b}.
\]

Therefore,

\[
(2^{2\ell} + 1) \text{rem}(w'_k - w''_k, b) + w''_k \equiv 1 \cdot (w'_k - w''_k) + w''_k \equiv w'_k \pmod{b}.
\]

\( \square \)
\[ w_k = (2^{2\ell} + 1) \text{rem}(w'_k - w''_k, b) + w''_k \]

Thus, \( w_k \) can be computed from \( w'_k \) and \( w''_k \) in \( O(\ell) \) bit operations (how?).

Another \( O(\ell) \) bit operations are needed to put \( w_k \) in its place (what does it mean?).

Summing up, we see that all \( w_k \)'s can be computed from the \( w'_k \)'s and \( w''_k \)'s in
\[ bO(\ell) = O(b\ell) = O(n) \]

bit operations.
Computation of $w'_k$
\[ w'_k = \text{rem} \left( \sum_{i+j=k} u_i v_j - \sum_{i+j=k+b} u_i v_j, b \right) = \text{rem} \left( \sum_{i+j=k} u'_i v'_j - \sum_{i+j=k+b} u'_i v'_j, b \right), \]

where
\[ u'_i = \text{rem}(u_i, b) \quad \text{and} \quad v'_j = \text{rem}(v_j, b). \]

Let
\[ y'_k = \sum_{i+j=k} u'_i v'_j. \]

Then
\[ w'_k = \text{rem}(y'_k - y'_{k+b}, b). \]

**Remark** Since \( u'_k, v'_k < b \) (why?), it follows that \( y'_k < b^3 \) (how?).
Let
\[ \hat{u} = \sum_{i=0}^{b-1} u'_i 2^{3i} \log_2 b \quad \text{and} \quad \hat{v} = \sum_{j=0}^{b-1} v'_j 2^{3j} \log_2 b. \]

Since \( y'_k < b^3 \), the \( y'_k \)'s are “mutually disjoint” in \( \hat{u} \hat{v} \). Therefore, we can “extract” all \( y'_k \)'s from the product in \( \hat{u} \hat{v} \).
Denote by $||m||$ the length of an integer $m$.

Since

$$||\hat{u}||, ||\hat{v}|| \leq ||3b \log_2 b||,$$

(why?), we can compute the product $\hat{u} \hat{v}$ in

$$O\left((b \log b)^{1.59}\right) = O\left((\sqrt{n \log n})^{1.59}\right) = O(n)$$

bit operations.
Computation of $w''_k$
Since
\[ u_i, v_j, w''_k < 2^{2\ell} + 1, \]
the computation will be performed modulo \( 2^{2\ell} + 1 \), i.e., in \( \mathbb{Z}_{2^{2\ell} + 1} \).

Recall that we have to compute the coefficients of \( \text{rem}(f(\lambda)g(\lambda), \lambda^b + 1) \).

Let \( \omega = 2^{2\ell}/b \). Then
\[ \omega^b = 2^{2\ell} \equiv -1 \mod 2^{2\ell} + 1, \]
implying
\[ \omega^{2b} \equiv 1 \mod 2^{2\ell} + 1. \]

Let \( s \) be such that \( b = 2^s \) (why there is such an \( s \))? Then
\[ b^{-1} \equiv 2^{4\ell-s} \mod 2^{2\ell} + 1, \]
because
\[ b \cdot 2^{4\ell-s} = (2^{2\ell})^2 \equiv_{2^{2\ell}+1} (-1)^2 = 1. \]
As we have already seen (where?),

\[
\begin{pmatrix}
  w_0'' \\
  \omega w_1'' \\
  \vdots \\
  \omega^i w_i'' \\
  \vdots \\
  \omega^{b-1} w_{b-1}''
\end{pmatrix}
= \frac{W(\omega^{2(b-1)})}{b}
\begin{pmatrix}
  w_0'' \\
  \omega w_1'' \\
  \vdots \\
  \omega^i w_i'' \\
  \vdots \\
  \omega^{b-1} w_{b-1}''
\end{pmatrix}
\cdot
\begin{pmatrix}
  u_0 \\
  \omega u_1 \\
  \vdots \\
  \omega^i u_i \\
  \vdots \\
  \omega^{b-1} u_{b-1}
\end{pmatrix}
\cdot
\begin{pmatrix}
  v_0 \\
  \omega v_1 \\
  \vdots \\
  \omega^i v_i \\
  \vdots \\
  \omega^{b-1} v_{b-1}
\end{pmatrix}.
\]
• Addition/subtraction in \( \mathbb{Z}_{2^{2\ell} + 1} \) can be performed in \( O(\ell) \) bit operations (why?).

• Multiplication by \( \omega^i \) is a shift followed by subtraction (how?) and, therefore can be also performed in \( O(\ell) \) bit operations.

• Therefore, since the Fourier transform can be preformed in \( O(b \log b) \) additions/subtractions and shifts, the \( w'''_{k} \)'s can be computed in \( O(\ell b \log b) + bM(2\ell) \), where \( M(n) \) is the number of bit operation needed to compute the product of two \( n \)-bit integers modulo \( 2^n + 1 \) (why?).

Therefore,
\[
M(n) \leq O(\ell b \log b) + bM(2\ell) + O(n)
= O(n \log n) + bM(2\ell) + O(n),
\]
where \( O(n) \) includes

• Computation of \( uv \) from \( w_{k} \)'s,

• computation of \( w'_{k} \)'s and \( w'''_{k} \)'s, and

• computation of \( w_{k} \)'s from \( w'_{k} \)'s and \( w'''_{k} \)'s

(why?).
\[ M(n) \leq cn \log n + bM(2\ell) \]

or

\[
\frac{M(n)}{n \log n} \leq c + \frac{M(2\ell)}{\ell \log n}
\]

(why?).

Since \( b = \ell \), if \( m \) is even, and \( 2b = \ell \), if \( m \) is odd (what is \( m \)?),

\[
\sqrt{n} \leq \ell \leq 2\sqrt{n}.
\]

If we denote \( \frac{3M(x)}{x \log x} \) by \( M'(x) \), then

\[
M'(n) \leq c + M'(4\sqrt{n})
\]

(why?).
\[ M'(n) \leq c + M'(4\sqrt{n}) \]

**Proposition**  For all \( t \geq 1 \),

\[ M'(n) \leq tc + M'(16n^{1/2t}) . \]

**Proof** The proof is by induction on \( t \). The basis is immediate (why?) and for the induction step assume that the inequality holds for \( t \). Then

\[
M'(n) \leq tc + M' \left( 16n^{1/2t} \right) \\
\leq tc + c + M' \left( 4 \left( 16n^{1/2t} \right)^{1/2} \right) \\
= (t + 1)c + M' \left( 16n^{1/2t+1} \right) .
\]

\[ \square \]
If \( t = \log_2 \log_2 n \), then

\[
n^{1/2^t} = n^{\frac{1}{\log_2 n}} = \left(2^{\log_2 n}\right)^{\frac{1}{\log_2 n}} = 2.
\]

Therefore,

\[
M'(n) \leq c \log \log n + M'(32),
\]

implying

\[
M(n) = O(n \log n \log \log n).
\]
Summary:

\[
uv \equiv \sum_{k=0}^{b-1} w_k 2^\ell k \mod 2^n + 1
\]

\[
w' = \text{rem}(w_k, b)
\]

\[
w'' = \text{rem}(w_k, 2^{2\ell} + 1)
\]

\[
w_k = (2^{2\ell} + 1)\text{rem}(w'_k - w''_k, b) + w''_k
\]