EXPONENTIATION

Problem: *Given two positive integers n and k, compute n^k.*

Straightforward algorithm:

Input: n, k (two positive integers)

Output: P (the value of n^k)

begin

$P \leftarrow n$;

for $i \leftarrow 1$ **to** $k - 1$ **do**

$P \leftarrow n \times P$;

end

IDEA: $n^k = n \times n^{k-1}$.

The first attempt algorithm requires k iterations. Since the size of k is $\log_2 k$, the number of iterations is exponential:

$$k = 2^{\log_2 k}.$$

How to reduce the number of iterations ?

IDEA: $n^k = \left(n^{\frac{k}{2}}\right)^2$.

Binary Algorithm:

Input: n, k (two positive integers)

Output: P (the value of n^k)

begin

$P \leftarrow 1$;

while $k \geq 1$ **do**

if $k \bmod 2 = 0$ **then begin** $n \leftarrow n \times n; k \leftarrow \frac{k}{2}$ **end**

else begin $P \leftarrow P \times n$; $k \leftarrow k - 1$ **end**

return P

end

The invariant is the product

$$P \times n^k.$$

Complexity:

$$2\log_2 k \in O(\log_2 k)$$

What about changing 2 to 3, $\ldots$, or to $t > 2$?

POLYNOMIAL MULTIPLICATION

Problem: *Compute the product of two given polynomials of degree* $n-1$.

$$P = \sum_{i=0}^{n-1} p_i x^i, \quad Q = \sum_{i=0}^{n-1} q_i x^i$$

$$\begin{aligned} PQ \quad &= \quad p_{n-1}q_{n-1}x^{2n-2} + \cdots \\ &\quad +(p_{n-1}q_{i+1} + p_{n-2}q_{i+2} + \cdots \\ &\quad +P_{i+1}q_{n-1})x^{n+i} + \cdots + p_0 q_0. \end{aligned}$$

The polynomial PQ has degree $2n-2$ and the coefficient of x^t is

$$\sum_{0\le i,j\le n, i+j=t} p_i q_j = p_0 q_t + p_1 q_{t-1} + p_2 q_{t-2} +$$

$$\cdots + p_i q_{t-i} + \cdots + p_t q_0.$$

How many operations?

$O(n^2)$ multiplications and additions.

DIVIDE–AND–CONQUER ALGORITHM

Assume that n is a power of 2. Divide each polynomial into two equal–sized parts:

$$P = P_1 + x^{\frac{n}{2}} P_2,$$

$$Q = Q_1 + x^{\frac{n}{2}} Q_2,$$

where

$$P_1 = p_0 + p_1 x + \cdots + p_{\frac{n}{2}-1} x^{\frac{n}{2}-1},$$

$$Q_1 = q_0 + q_1 x + \cdots + q_{\frac{n}{2}-1} x^{\frac{n}{2}-1},$$

$$P_2 = p_{\frac{n}{2}} + p_{\frac{n}{2}+1}x + \cdots + p_{n-1}x^{\frac{n}{2}-1},$$

$$Q_2 = q_{\frac{n}{2}} + q_{\frac{n}{2}+1}x + \cdots + q_{n-1}x^{\frac{n}{2}-1}.$$

So,

$$\begin{aligned} PQ &= \left(P_1 + P_2x^{\frac{n}{2}}\right)\left(Q_1 + Q_2x^{\frac{n}{2}}\right) \\ &= P_1Q_1 + (P_1Q_2 + P_2Q_1)x^{\frac{n}{2}} + P_2Q_2x^n. \end{aligned}$$

Remark: PQ involves products of polynomials of degree $\frac{n}{2}$!

Compute the product of the smaller polynomials (e.g. P_1Q_1) then add the results to complete the solution.

Constraints:

- smaller problems should be exactly as the original problem,
- we know how to multiply polynomials of degree 1.

BOTH CONDITIONS ARE ACTUALLY SATISFIED!

Let $T(n) =$ be the number of operations:

$$\begin{cases} T(1) = 1, \\ T(n) = 4T(\frac{n}{2}) + O(n). \end{cases}$$

Here the factor 4 comes from the products of smaller polynomials.

So,

$$T(n) \in O(n^2),$$

so no improvement has been obtained !

Rewrite the formula

$$PQ = P_1Q_1 + (P_1Q_2 + P_2Q_1)x^{\frac{n}{2}} + P_2Q_2x^n$$

in the form

$$\begin{array}{ccc} x & P_1 & P_2 \\ Q_1 & A & B \\ Q_2 & C & D \end{array}$$

and compute

$$A + (B + C)x^{\frac{n}{2}} + Dx^n.$$

Remark: We do not have to compute B and C separately as we need only their sum $B + C$.

Write:

$$E = (P_1 + P_2)(Q_1 + Q_2),$$

so

$$B + C = E - A - D.$$

Hence, we need to compute only A, D, E. All the rest (additions, subtractions) contribute only $O(n)$.

The new recurrence relation is:

$$T(n) = 3T(\frac{n}{2}) + O(n),$$

so,

$$T(n) \in O\left(n^{\log_2 3}\right) = O(n^{1.59}).$$

Example:

$$P = 1 - x + 2x^2 - x^3$$
$$Q = 2 + x - x^2 + 2x^3$$

In the straightforward algorithm we use 16 multiplications and 9 additions and subtractions. By divide-and-conquer we get:

$$A = (1 - x)(2 + x) = 2 - x - x^2,$$

$$D = (2 - x)(-1 + 2x) = -2 + 5x - 2x^2,$$

$$E = (3 - 2x)(1 + 3x) = 3 + 7x - 6x^2,$$

$$B + C = E - A - D = 3 + 3x - 3x^2$$

$$PQ = A + (B + C)x^2 + Dx^4$$

in 12 multiplications + 13 additions/subtractions.

MATRIX MULTIPLICATION

$$A = (a_{ij}), \quad B = (b_{ij}) \quad i, j = 1, 2 \ldots, n$$

$$C = AB,$$

$$C = (c_{ij})$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \times b_{kj}$$

The straightforward way to compute C requires n^3 multiplications and $(n-1)n^2$ additions.

Winograd's Algorithm

Assume n is even and put

$$A_i = \sum_{k=1}^{\frac{n}{2}} a_{i,2k-1} \times a_{i,2k},$$

$$B_j = \sum_{k=1}^{\frac{n}{2}} b_{2k-1,j} \times b_{2k,j}$$

Re-arrange terms and get:

$$c_{ij} = \sum_{k=1}^{\frac{n}{2}} (a_{i,2k-1} + b_{2k,j})(a_{i,2k} + b_{2k-1,j}) - A_i - B_j$$

Indeed,

$\sum_{k=1}^{\frac{n}{2}}(a_{i,2k-1} + b_{2k,j})(a_{i,2k} + b_{2k-1,j}) - A_i - B_j =$

$\sum_{k=1}^{\frac{n}{2}}(a_{i,2k-1} + b_{2k,j})(a_{i,2k} + b_{2k-1,j}) - a_{i,2k-1}a_{i,2k} - b_{2k-1,j}b_{2k,j} =$

$\sum_{k=1}^{\frac{n}{2}} a_{i,2k-1}b_{2k-1,j} + b_{2k,j}a_{i,2k} = \sum_{k=1}^{n} a_{ik}b_{kj}.$

Computation of A_is and B_is requires n^2 multiplications, so globally one uses

$$\frac{1}{2}n^3 + n^2$$

multiplications. Is it an improvement? Yes, in case additions can be performed much faster than multiplications.

STRASSEN's ALGORITHM

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},$$

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

$$C = AB,$$

so by straightforward computation we perform 8 multiplications, and 4 additions.

Divide-and-conquer approach:

Let n be a power of 2,

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},$$

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix},$$

where a_{ij}s, b_{ij}s,c_{ij}s are $\frac{n}{2} \times \frac{n}{2}$ matrices. Treat these submatrices as *elements*: the algorithm for 2×2 matrices can be converted to an $n \times n$ product ...

How ? By substituting a recursive call each time a product of elements appears.

We get the recurrence:

$$T(n) = 8T\left(\frac{n}{2}\right) + O\left(n^2\right),$$

which implies

$$T(n) \in O\left(n^{\log_2 8}\right) = O\left(n^3\right),$$

for the straightforward algorithm.

Question: Can we do it better for 2×2 products ?

Strassen's Method

$$
\begin{aligned}
P_1 &= (a_{11} + a_{22})(b_{11} + b_{22}), \\
P_2 &= (a_{21} + a_{22})b_{11}, \\
P_3 &= a_{11}(b_{12} - b_{22}), \\
P_4 &= a_{22}(-b_{11} + b_{21}), \\
P_5 &= (a_{11} + a_{12})b_{22}, \\
P_6 &= (-a_{11} + a_{21})(b_{11} + b_{22}), \\
P_7 &= (a_{12} - a_{22})(b_{21} + b_{12}),
\end{aligned}
$$

$$
\begin{aligned}
c_{11} &= P_1 + P_4 - P_5 + P_7, \\
c_{12} &= P_3 + P_5, \\
c_{21} &= P_2 + P_4, \\
c_{22} &= P_1 + P_3 - P_2 + P_6,
\end{aligned}
$$

18 additions/multiplications, but 7 multiplications !

Strassen's algorithm complexity:

$$T(n) \in O\left(n^{\log_2 7}\right) = O\left(n^{2.81}\right).$$

A Better Method: 7 multiplications and 16 additions

$r = a_{21} + a_{22}$, $t = r - a_{11}$,

$q = b_{22} - b_{12}$,

$m_1 = t(q + b_{11})$,

$m_2 = a_{11}b_{11}, m_3 = a_{12}b_{21}$,

$m_4 = (a_{11} - b_{21})q$,

$m_5 = q(b_{12} - b_{11})$,

$m_6 = (a_{12} - t)b_{22}$,

$m_7 = a_{22}(b_{11} - b_{21} + q)$,

$c_{11} = m_2 + m_3$,

$c_{12} = p + m_5 + m_6$,

$p = m_1 + m_2$, $s = p + m_4$,

$c_{21} = s - m_7$,

$c_{22} = s + m_5$,

LINEAR REPRESENTATIONS

Let $n \geq 1$, $\mathcal{Z}$ be the set of non-zero integers and

$$A = \{a_1, a_2, \ldots, a_n\} \subset \mathcal{Z}.$$

Consider the set of all linear combinations of elements of A:

$$M(A) = \left\{ \sum_{i=1}^{n} x_i a_i | x_i \in \mathcal{Z} \right\}.$$

Problem: Given $m \in \mathcal{Z}$, test whether $m \in M(A)$?

Main difficulty:

$$m \in M(A) \Longleftrightarrow \exists x_1, \ldots, x_n \in \mathcal{Z} :$$

$$m = x_1 a_1 + x_2 a_2 + \cdots + x_n a_n.$$

The above test is *infinite*!

Mathematical Fact:

$M(A)$ is closed under *difference*

that is:

$$s, t \in M(A) \Longrightarrow s - t \in M(A).$$

Indeed,

$$\begin{aligned} s &= \sum_{i=1}^{n} x_i a_i, \\ t &= \sum_{i=1}^{n} y_i a_i, \\ s - t &= \sum_{i=1}^{n} (x_i - y_i) a_i. \end{aligned}$$

Accordingly,

$$(M(A), +)$$

is a group under addition, more, a subgroup of the additive group of integers

$$(\mathcal{Z}, +) \tag{$*$}$$

But, every subgroup of $(*)$ is of the form

$$q\mathcal{Z} = \{qx \mid x \in \mathcal{Z}\},$$

where $q \geq 0$.

So, there exists $q \in \mathcal{Z}$ such that $M(A) = q\mathcal{Z}$.

$I. \quad q = GCD(a_1, a_2, \ldots, a_n)$

$II. \quad m \in M(A) \Longleftrightarrow m \in q\mathcal{Z} \Longleftrightarrow GCD(a_1, a_2, \ldots, a_n) | m.$

q is called a "finite certificate"

TESTING IRREDUCIBILITY

A) *Testing primality for naturals is theoretically easy, but it appears to be difficult to do it faster !*

Test the condition

$$x|N$$

up to $\lceil\sqrt{N}\rceil$.

This is $O(\sqrt{N})$, i.e. an exponential algorithm.

CAN WE DO IT BETTER ?

B) A polynomial

$$P \in Q[x]$$

is *irreducible* if there are no polynomials $S, T \in Q[x]$ with degrees ≥ 1 such that

$$P = ST$$

Problem: Given a polynomial P with rational coefficients, test whether P is irreducible.

IDEA: Reduce the problem to polynomials with integer coefficients.

Ingredient:

Gauss' Lemma: *The product of two primitive polynomials with integer coefficients is a primitive polynomial.*

Recall that a polynomial

$$f = a_0 + a_1 X + \cdots + a_n X^n$$

is primitive if

$$GCD(a_0, a_1, \ldots, a_n) = 1.$$

Proof: Let

$$f = a_0 + a_1 X + \cdots + a_n X^n,$$

$$g = b_0 + b_1 X + \cdots + b_m X^m,$$

two primitive polynomials in $\mathcal{Z}[X]$ and let

$$fg = c_0 + c_1 X + \cdots + c_{n+m} X^{n+m}.$$

Let p be a prime.

As $GCD(a_0, a_1, \ldots, a_n) = 1$ we can get the smallest k such that

$$p | a_0, \ldots, p | a_{k-1} \text{ but } p \not| a_k.$$

Similarly, as

$$GCD(b_0, b_1, \ldots, b_m) = 1$$

we construct the smallest l such that

$p|b_0, p|b_1, \ldots, p|b_{l-1}$, but

$$p \not| b_l.$$

The coefficient of

$$X^{k+l}$$

in fg is

$$c_{k+l} = \sum_{0 \le i \le n, 0 \le j \le m, i+j=k+l} a_i b_j$$

and $p \not| c_{k+l}$ (as $p \not| a_k b_l$, but p divides all other terms).

So, $p \not| GCD(c_0, \ldots, c_{m+n})$.

Let $f \in \mathcal{Q}[X]$ be a polynomial of degree $n > 1$. Assume

$$f = qg,$$

where $g \in \mathcal{Z}[X]$ is *primitive* and $q \in \mathcal{Q}$.

Compute

$$g(0), g(1), \ldots, g(n).$$

If $g(m) = 0$, for some $0 \leq m \leq n$, then $g = (X - m)h$, so f *is not* irreducible.

Assume now that

$$g(i) \neq 0,\ 0 \leq i \leq n.$$

Reducing the infinite search to an equivalent, finite one:

For every function

$$\alpha : \{0, 1, \ldots, n\} \longrightarrow \mathcal{Z}$$

such that $\alpha(m)$ is a divisor of $g(m)$, for $m = 0, 1, \ldots, n$
construct the *unique polynomial*

$$g_\alpha \in \mathcal{Z}[X]$$

of degree $\leq n$, such that

$$(*) \qquad g_\alpha(m) = \alpha(m), \; m = 0, 1, \ldots, n.$$

Indeed, let

$$g_\alpha = d_0 + d_1 X + \cdots + d_n X^n.$$

From $(*)$ we deduce the relations:

$$\begin{array}{rcl} d_1 + d_2 + \cdots + d_n & = & \alpha(1) - \alpha(0), \\ 2d_1 + 2^2 d_2 + \cdots + 2^n d_n & = & \alpha(2) - \alpha(0), \\ & \vdots & \\ nd_1 + n^2 d_2 + \cdots + n^n d_n & = & \alpha(n) - \alpha(0), \end{array}$$

a $n \times n$ system with the determinant

$$1!\,2!\ldots n! \neq 0.$$

We have got the finite set

$$F = \{g_\alpha \in \mathcal{Z}[X] | \alpha : \{0, 1, \ldots, n\} \to \mathcal{Z}, \forall\, 0 \leq m \leq n,$$
$$g_\alpha(m) = \alpha(m) | g(m)\}.$$

Lemma: If $g = h_1 h_2, h_i \in \mathcal{Z}[X]$, then $h_i \in F$.

Proof: If $g(m) = h_1(m) h_2(m) \neq 0$, $0 \leq m \leq n$, so

$$h_i(m) | g(m) \qquad i = 1, 2$$

$h_i = g_{\alpha_i}$, where $\alpha_i(m) = h_i(m)$, $0 \leq m \leq n$.

Example:

$$g = 1 \times g$$

$1 \in F$ corresponds to $\alpha(i) = 1$, $0 \leq i \leq n$,
$g \in F$ corresponds to $\alpha(i) = g(i)$, $0 \leq i \leq n$.

In particular,

$$F \supset \{1, g\}.$$

Final Lemma: If

$$f = GH, \quad G, H \in \mathcal{Q}[X],$$

then there exist (and can be effectively computed)

$$q_G, q_H \in \mathcal{Q},$$

$$h_G, h_H \in F$$

such that

$$G = q_G h_G, \quad H = q_H h_H.$$

Proof: Routine computation gives

$$f = GH = (q_G h_G)(q_H h_H) = (q_G q_H)(h_G h_H).$$

The polynomials h_G, h_H are primitive, so by Gauss Lemma, their product is primitive, and

$$f = \underbrace{(q_G q_H)}_{\in \mathcal{Q}} \underbrace{(h_G h_H)}_{\in \mathcal{Z}[X]} = qg,$$

$$q = q_G q_H, \quad g = h_G h_H.$$

By Lemma, $h_H, h_G \in F$.

We have proved :

Kronecker's Theorem: *There exists an algorithm for testing if an arbitrary polynomial* $f \in \mathcal{Q}[X]$ *is irreducible.*

The finite set F is a "finite certificate".

Example

$$\begin{aligned} f &= \frac{1}{66}X^3 + \frac{1}{11}x^2 + \frac{1}{6}X + \frac{1}{11} \\ &= \frac{1}{66}\left(X^3 + 6X^2 + 11X + 6\right) \end{aligned}$$

$g = X^3 + 6X^2 + 11X + 6$

is primitive as

$$GCD(1, 6, 11, 6) = 1.$$

We continue with g:

$g(0) = 6,$
$g(1) = 1 + 6 + 11 + 6 = 24,$
$g(2) = 8 + 6 \times 4 + 11 \times 2 + 6 = 60,$
$g(3) = 27 + 6 \times 9 + 11 \times 3 + 6 = 120.$

$\alpha : \{0, 1, 2, 3\} \longrightarrow \mathcal{Z}$
$\alpha(0) \mid g(0) = 6,$
$\alpha(1) \mid g(1) = 24,$

$\alpha(2) \mid g(2) = 60,$
$\alpha(3) \mid g(3) = 120.$

Consider:

$$\alpha(0) = 3,$$

$$\alpha(1) = 4,$$

$$\alpha(2) = 5,$$

$$\alpha(3) = 6,$$

Clearly, α satisfies the above restrictions !

Let

$$g_\alpha = a_0 + a_1 X + a_2 X^2 + a_3 X^3$$

So, from the relations

$$g_\alpha(0) = \alpha(0),$$

$$g_\alpha(1) = \alpha(1),$$

$$g_\alpha(2) = \alpha(2),$$

$$g_\alpha(3) = \alpha(3),$$

we deduce the system:

$$\begin{cases} a_0 = \alpha(0), \\ a_0 + a_1 + a_2 + a_3 = \alpha(1), \\ a_0 + 2a_1 + 4a_2 + 8a_3 = \alpha(2), \\ a_0 + 3a_1 + 9a_2 + 27a_3 = \alpha(3), \end{cases}$$

that is

$$\begin{cases} a_1 + a_2 + a_3 = 1, \\ 2a_1 + 4a_2 + 8a_3 = 2, \\ 3a_1 + 9a_2 + 27a_3 = 3, \end{cases}$$

The system has a unique solution: $a_1 = 1, a_2 = 0, a_3 = 0$ ($a_0 = \alpha(0) = 3$).

So,

$$g_\alpha = 3 + X$$

and

$$g = g_\alpha \times \left(X^2 + 3X + 2\right),$$

$$f = \frac{1}{66}\,(3 + X)\left(X^2 + 3X + 2\right)$$

is not irreducible !