TUTORIAL 2

1 Multiplication of two polynomials

Give an algorithm to multiply a degree 1 polynomial by a degree 2 polynomial in at most 4 multiplications.

2 Alternative FFT algorithm

Let P be a polynomial of degree at most $2^k - 1$, and write $P = P_h X^{2^{k-1}} + P_l$. Let ω be a primitive 2^k-th root of 1.

1. Prove that $P(\omega^{2i}) = P_h(\omega^{2i}) + P_l(\omega^{2i})$ and $P(\omega^{2i+1}) = -P_h(\omega^{2i+1}) + P_l(\omega^{2i+1})$.

2. Deduce an alternative FFT algorithm. (Hint: introduce the polynomial $Q(X) = P_l(\omega X) - P_h(\omega X)$).

3 Is squaring easier than multiplying?

Show that computing the square of a n-digit number is not (asymptotically) easier than multiplying two n-digit numbers.

4 The "binary splitting" method computation of $n!$

We want to compute $n!$ and assume that n is a "small" integer (i.e. it fits into one machine word). We denote with $M(k)$ the cost (in terms of elementary operations) of the multiplication of two k-bit numbers, and we assume $2M(k/2) \leqslant M(k)$ (we remind some typical values: $M(k) = O(k^2)$ with naive multiplication, $O(k^{\log(3)/\log(2)})$ with Karatsuba multiplication and $O(k \log k \log\log k)$ with the FFT-in finite ring variant of the Schönhage & Strassen algorithm). Use the fact that $\log n! \sim n \log n$.

1. What is the cost of multiplying $O(n)$-digit integer by a $O(1)$-digit integer by the naive algorithm. Argue that it is essentially optimal.

2. We first consider the simplest approach: $x_1 = 1,\ x_2 = 2x_1,\ x_3 = 3x_2,\ \ldots,\ x_n = nx_{n-1}$. Show that the cost of this approach is $O(n^2(\log n)^2)$.

3. We define
$$p(a, b) = (a + 1)(a + 2)\cdots(b - 1)b = \frac{b!}{a!}.$$
Suggest a recursive method to compute $n!$ with cost $O(\log n M(n \log n))$. With the classical multiplication algorithms, is this more interesting than the simple method?

5 Recursive division

Let a and b be two polynomials in $K[x]$ such that $\deg a = 4n$ and $\deg b = 2n$ and take n to be a power of 2. We decompose a and b such that $a(x) = a_h(x)x^{2n} + a_l(x)$ and $b(x) = b_h(x)x^n + b_l(x)$, where $\deg a_h, \deg a_l \leqslant 2n$ and $\deg b_h, \deg b_l \leqslant n$.

Consider $D(n)$ as the complexity, in number of arithmetic operations over K, required to perform the euclidean division of a degree $2n$ polynomial by a degree n polynomial. Similarly, we denote by $M(n)$ the complexity of multiplying two degree n polynomials over R.

We perform the euclidean division of a_h by b_h (i.e. $a_h = b_h q_h + r_h, \deg r_h < \deg b_h$).

1. Show that $\deg(a - bq_h x^n) < 3n$ and that $a - bq_h x^n$ is computable using $D(n) + M(n) + O(n)$ operations.

2. Show that we can finish dividing a by b using another $D(n) + M(n) + O(n)$ operations.

3. What is the value of $D(n)$ if $M(n) = n^\alpha, \alpha > 1$?

4. Same question as before, for $M(n) = n(\log n)^\alpha, \alpha > 1$.