TUTORIAL 2
05.02.19

1 Is squaring easier than multiplying?

Show that computing the square of an n-digit number is not (asymptotically) easier than multiplying two n-digit numbers. We assume we work in a ring where we can divide by 2.

2 Exact division

The goal of this exercise is to show that we can have a constant gain on (naive) polynomial division in the case where we know beforehand that the division is exact. As is always the case with constant gain, we will need to argue in the end that our complexity model is sound. Let $A(X) = \sum_{k=0}^{2n-1} a_k X^k$ and $B(X) = \sum_{k=0}^{n-1} b_k X^k$.

1. Prove that one can compute $Q_\ell = \sum_{k=n-\ell}^{n} q_k X^k$ such that $\deg(A - BQ_\ell) < 2n - 1 - \ell$ using $\ell + 1$ divisions and $\ell(\ell+1)/2$ multiplications in K. *Note that we are not interested in the remainder $A - BQ_\ell$, only in Q_ℓ.*

2. Prove that the algorithm of **1**. can be used to compute $S_\ell = \sum_{k=0}^{\ell} s_k X^k$ such that $\mathrm{val}(A - S_\ell B) > \ell$ using $\ell + 1$ divisions and $\ell(\ell+1)/2$ multiplications in K. (Where $\mathrm{val}(P)$ is the smallest degree of monomial of P with nonzero coefficient.)

3. Assume that we know, for some reason, that $B|A$ and want to compute A/B. Use **1** & **2** to give an algorithm for this task, and compare this with the "schoolbook" division.

4. We only counted divisions and multiplications, but in a standard algebraic model, addition and subtraction also have a comparable cost. Does our result really make sense?

3 Multiplication of bivariate polynomials

Fact: Let $c_0, \dots c_d$ be $d + 1$ distinct elements of K and $Q_0, \dots Q_d \in K[X]$. There is a unique polynomial $P \in K[X, Y]$ of Y-degree at most d satisfying $P(X, c_i) = Q_i$ for every $i = 0, \dots d$.

Let us assume that we can efficiently find such P. Again, assume that operations in K have unit cost.

1. What is the cost of a naive multiplication of two bivariate polynomials A and B of X-degree at most D_1 and Y-degree at most D_2?

2. Give an algorithm that computes $A(X, c)$ for a given $c \in K$, with A of X-degree at most D_1 and Y-degree at most D_2. What is its cost?

3. Assuming that $|K| \geq 2D_2 + 1$ and using the fact above, describe an algorithm for multiplying bivariate polynomials (which would, assuming that we have a fast algorithm for multiplication of polynomials of one variable, beat the naive multiplication).

4 Polynomial Translation

Let R denote a ring, a an element of R, and P a polynomial over R of degree at most n. We want to compute the polynomial $Q(X) = P(X + a)$.

1. Propose a direct algorithm to compute Q and estimate its complexity (as a function of n).
2. Prove that it is possible to compute $Q(X)$ with only $O(M(n) \log n)$ operations $(+, -, \times)$ in R, where $M(n)$ denotes the complexity of degree-n polynomial multiplication.