Computational Algebra. Lecture 9

M.Eulàlia Montoro

20 de abril de 2022

Schönage & Strassen

- Input: $f, g \in A[X]$ of $\deg < n = 2^k$, (2 is a unit in A)
- Output: $h \in A[X]$ such that $fg \equiv h \bmod (X^n + 1)$, $\deg h < n$.
-
  1. If $k \leq 2$ then compute $f.g$ and return $f.g \operatorname{rem} (X^n + 1)$
  2. $2^{\lfloor k/2 \rfloor} \mapsto m$, $n/m \mapsto t$
     Let $f', g' \in A[X, Y]$ with $\deg_X f'$, $\deg_X g' < m$ such that $f = f'(X, X^m)$ and $g = g'(X, X^m)$
  3. Let $D = A[X]/\langle X^{2m} + 1 \rangle$
     If $t = 2m$ then $\eta \mapsto X \bmod (X^{2m} + 1)$ else $\eta \mapsto X^2 \bmod (X^{2m} + 1)$ (η is a primitive $2t - th$ root of unity)
     $f'' \mapsto f' \bmod (X^{2m} + 1)$, $g'' \mapsto g' \bmod (X^{2m} + 1)$
     Call the fast convolution algorithm with $\omega = \eta^2$ to compute $h'' \in D[Y]$ of degree less than t such that $f''(\eta Y)g''(\eta Y) \equiv h''(\eta Y) \bmod (Y^t - 1)$
  4. Let $h' \in A[X, Y]$ with $\deg_X h' < 2m$ such that $h'' = h' \bmod (X^{2m} + 1)$
     $h'(X, X^m) \operatorname{rem} (X^n + 1) \mapsto h$, Return h.

Homework assignment

Implement the algorithm above in $A = \mathbb{F}_p$, with $p > 2$ prime. Test its speed with random examples.

Homework assignment (Hint)

2. Let $f', g' \in A[X, Y]$ with $\deg_X f'$, $\deg_X g' < m$ such that $f = f'(X, X^m)$ and $g = g'(X, X^m)$

(* We obtain f' and g' *)

$$f2 = PolynomialRemainder[f, y - x^m, x];$$

$$g2 = PolynomialRemainder[g, y - x^m, x];$$

Homework assignment (Hint)

```
(* Step 3 *)
η = If[t == 2 m, PolynomialMod[x, x ^ (2 * m) + 1], PolynomialMod[x ^ 2, x ^ (2 * m) + 1]];
     si            función mod polinómica                    función mod polinómica
P3 = PolynomialMod[P2, x ^ (2 * m) + 1];
     función mod polinómica
Q3 = PolynomialMod[Q2, x ^ (2 * m) + 1];
     función mod polinómica
P4 = P3 /. y → η * y;
Q4 = Q3 /. y → η * y;

H1 = PFFT[CoefficientList[P4, y], CoefficientList[Q4, y], 2 * t];
          lista de coeficientes   lista de coeficientes
H2 = FromDigits[Expand[Reverse[Chop[Collect[H1, x]]]], Modulus → p], y];
     de dígitos a n⋯ expande⋯ invierte o⋯ cam⋯ agrupa coeficientes  módulo
H3 = PolynomialMod[Expand[H2 /. y → η^-1 * y], x ^ (2 * m) + 1];
     función mod polinó⋯ expande factores
```