MATH 895, Course Project, Summer 2019

Black Boxes and Zippel's Sparse Interpolation

Instructor: Michael Monagan

The project is worth 40% of your final grade.

Let $f \in \mathbb{Z}[x_1, x_2, \ldots, x_n]$ be represented by a black box $\mathbf{B}$ such that given a prime p and an evaluation point $\alpha \in \mathbb{Z}_p^n$ the call $\mathbf{B}(\alpha, p)$ outputs $f(\alpha) \bmod p$. The goal of this project is to design collection of operations on black boxes and implement them in Maple. All of the algorithms will be probabilistic. We will pick p sufficiently large so that they work with high probability (w.h.p.). Implement the following routines in Maple.

1. $\mathbf{B}(\alpha,p)$ outputs $f(\alpha) \bmod p$.
2. **degBB**(B,n) outputs $\deg(f)$ the total degree of f w.h.p.
   degBB(B,n,i) outputs $\deg(f, x_i)$ the degree of f in x_i w.h.p.
   If $f = 0$ then output -1.
3. **suppBB**(B,$[x_1, \ldots, x_n]$) outputs the support of f i.e. the set of monomials of f w.h.p.
4. **sintBB**(B,$[x_1, \ldots, x_n]$) outputs the polynomial f w.h.p., i.e. interpolates f from the black box and recovers the integer coefficients of f using Chinese remaindering.

Notes

- To implement this in Maple the black box will be represented by a Maple procedure that computes f. It might look like this

  ```
  proc( alpha::list(integer), p::prime ) ...  end.
  ```

  A simple test example could be

  ```
  > B := proc(alpha::list(integer), p::prime) local f;
  >    f := 3*x^2-5*x*y*z+11*z^3;
  >    Eval(f,{x=alpha[1],y=alpha[2],z=alpha[3]} mod p;
  > end:
  ```

- To measure the efficiency of the algorithms, each of `degBB, suppBB, sintBB` should print out the number of calls to the black box **B** that it makes. To do this use a global counter like this

```
> B := proc(alpha::list(integer), p::prime)
>    global CNT;
>    CNT := CNT + 1;
>    ...
> end:
> CNT := 0; # don't forget to initialize it
```

- For the procedure **suppBB** use Zippel's sparse interpolation method. Use the `degBB` procedure to determine $\deg(f, x_i)$ for $1 \leq i \leq n$ first. For the sparse interpolation step, pick $\beta \in \mathbb{Z}_p^n$ at random and compute

$$f(\beta_1^j, \beta_2^j, \ldots, \beta_n^j) \bmod p \ \text{ for } \ 1 \leq j \leq t$$

  and solve the shifted Vandermonde system.

- Zippel's sparse interpolation method first evaluates the first variable x_1 at some point β_1 then interpolates $f(\beta_1, x_2, \ldots, x_n)$ recursively. After this is done it uses the support of $f(\beta_1, x_2, \ldots, x_n)$ to obtain $f(\beta_i, x_2, \ldots, x_n)$ for $i = 2, 3, \ldots, \deg(f, x_1)$. Given a blackbox **B** that computes $f(\alpha) \bmod p$, to create a black box **C** for the recursive call in one less variable, use

```
> beta1 := ...;
> C := proc(alpha::integer,p::prime)
>     B([beta1,op(alpha)],p)
> end:
> ...
```

  So the **C** procedure takes as input a list of $n-1$ values $[\alpha_1, \ldots, \alpha_{n-1}]$ for $x_2, ..., x_n$ and calls **B** with n values $[\beta, \alpha_1, \ldots, \alpha_{n-1}]$.

  After you have interpolated $g := f(\beta_1, x_2, \ldots, x_n) \bmod p$ you will have a Maple polynomial and you will need to get the support of g, that is, the monomials in g. Use the `coeffs` command like this

```
> C := coeffs(g,indets(g),'S');
> S := [S]; # support of g
```

- Suppose

$$S = \text{Support}(f(\beta_1, x_2, \ldots, x_n), \{x_2, \ldots, x_n\}) \text{ and } T = \text{Support}(f, \{x_2, \ldots, x_n\}).$$

So T is the true support and S is the support at $x_1 = \beta_1$. For example, if $f(x, y, z) = x^2yz - 4yz + 3xy$ then $T = \{yz, y\}$ and $S = \text{Support}(f(2, x, y)) = \{y\}$.

What is the probability that S is wrong, that is, $S \neq T$?
Use the Schwartz-Zippel Lemma to give a precise bound.

Design a probabilistic test to check if $S = T$?
What is the probability your test outputs true but $S \neq T$?

- For the procedure **sintBB** you need to first determine the support of f then solve for the coefficients of f. Use additional primes and Chinese remaindering to determine the coefficients. For each additional prime q assume the support obtained using the first prime p is the support of f so that you can use a Sparse Interpolation to solve for the coefficients mod q.

- Zippel's sparse interpolation method first evaluates the first variable x_1 at some point β_1 at random from $[0, p)$ then interpolates $f(\beta_1, x_2, \ldots, x_n)$ recursively. Then to compute $f(\beta_i, x_2, \ldots, x_n)$ for $i = 2, 3, \ldots$ we use sparse interpolation.

- We need a good application. For the application suppose we are given an $m \times m$ matrix A of polynomials in $\mathbb{Z}[x_1, ..., x_n]$ and let $f(x_1, \ldots, x_n) = \det A$. Program your black box **B** to compute $\det(A(\alpha_1, \ldots, \alpha_n) \bmod p) \bmod p$.

  Test your code working for the following three matrices. The first is a 3 by 3 matrix in three variables a, b, c. The second is a 6 by 6 matrix in 7 variables a, b, c, d, e, f, z. The third one forces you to use Chinese remaindering. The matrix is the same as the second but I've set $a = 11$ so the variables are b, c, d, e, f, z. Use 31 bit primes.

```
> with(LinearAlgebra):
> A := Toeplitz([a,b,d],symmetric);
```

$$A := \begin{bmatrix} a & b & c \\ b & a & b \\ c & b & a \end{bmatrix}$$

```
> A := Matrix(6,6,
  [[8*a^3, 4*a^4-4*a^2*d^2+4*a^2*f^2, 0, 0, -4*a^4+4*a^2*b^2-4*a^2*c^2, 0],
   [0, 8*a^3, 4*a^4-4*a^2*b^2+4*a^2*c^2, -8*a^3*c^2, 0, 0],
   [0, 0, 8*a^3, -4*a^4+4*a^2*b^2-4*a^2*c^2, 0, 0],
   [-4*a^4+4*a^2*d^2-4*a^2*f^2, -8*a^3*f^2, 0, 0,
      4*a^3*c^2-4*a^3*e^2+4*a^3*f^2, 48*a^2*z],
   [0, 0, -4*a^4+4*a^2*d^2-4*a^2*f^2, 4*a^3*c^2-4*a^3*e^2+4*a^3*f^2, -8*a^3, 0],
   [0, 0, 0, 48*a^2*z, 0, -8*a^3]]);
```

```
> a := 11; # same matrix as above
> A := Matrix(6,6,
  [[8*a^3, 4*a^4-4*a^2*d^2+4*a^2*f^2, 0, 0, -4*a^4+4*a^2*b^2-4*a^2*c^2, 0],
   [0, 8*a^3, 4*a^4-4*a^2*b^2+4*a^2*c^2, -8*a^3*c^2, 0, 0],
   [0, 0, 8*a^3, -4*a^4+4*a^2*b^2-4*a^2*c^2, 0, 0],
   [-4*a^4+4*a^2*d^2-4*a^2*f^2, -8*a^3*f^2, 0, 0,
      4*a^3*c^2-4*a^3*e^2+4*a^3*f^2, 48*a^2*z],
   [0, 0, -4*a^4+4*a^2*d^2-4*a^2*f^2, 4*a^3*c^2-4*a^3*e^2+4*a^3*f^2, -8*a^3, 0],
   [0, 0, 0, 48*a^2*z, 0, -8*a^3]]);
```

What to hand in?

To present your work please write a report in LaTeX. The report should be 10 to 12 pages (12pt font, 1 inch margins) plus any appendices that you wish to include. You should explain selected details of the algorithms and present any data and/or examples that you wish to show. Submit also a printout of a Maple worksheet showing your Maple code and demonstrating that the code works correctly.

Assessment

Code demo **(10 marks)** Monday July 29th
Final report **(15 marks)** and final code **(15 marks)** due 5pm August 16th.