Lec19B Handouts

March 23, 2021 1:51 PM

We are factoring polynomials in $\mathbb{Z}_p[x]$.

Ch 8 Polynomial Factorization

Let $a \in R[x]$, $d = \deg a$.

✓ 8.2 Square-free factorization $\mathbb{Q} \subset R$

× 8.3 Square-free factorization over Finite Fields

× 8.4 Berlekamps factorization algorithm for $\mathbb{Z}_p[x]$ (1967)

$O(d^3 + pd^2)$ arith. ops in $\mathbb{Z}_p$

Solve $Ax=0$ p gcds

× 8.5 The "big prime" Berlekamp algorithm for $\mathbb{Z}_p[x]$ (1970)

$O(d^3 + \log p \cdot d^2)$ arith. ops in $\mathbb{Z}_p$

✓ 8.6 Cantor-Zassenhaus algorithm for $\mathbb{Z}_p[x]$ (1981)

$O(d^3 \log p)$ arith. ops in $\mathbb{Z}_p$

✓ 8.7 Factorization in $\mathbb{Z}[x]$ (Hensel lifting) (1966)

× 8.8 Factorization in $\mathbb{Q}(\alpha)[x]$ (1976)

$\downarrow$ resultants

$\mathbb{Q}[x]$ + gcds in $\mathbb{Q}(\alpha)[x]$

with FFT

$O(d \, d \log d \log p) = O(d^2 \log d \log p)$.

```
> restart;
  p := prevprime(2^30);
```

$$p := 1073741789$$

```
> R := rand(p):
> d := 100;
```

$$d := 100$$

```
> f := 1:
  for i to d do
      alpha := R();
      while Rem(f,x-alpha,x) mod p = 0 do alpha := R() od;
      f := Expand( f*(x-alpha) ) mod p;
  od:
```

$f = \prod_{i=1}^{100} (x - \alpha_i)$ with 100 linear factors.

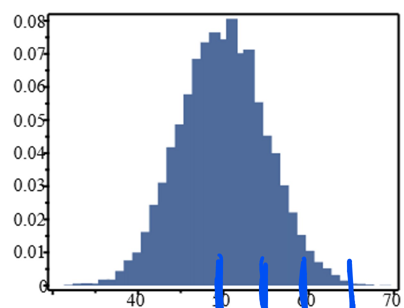
```
> N := 10000;
  F := Array(0..d):
  to N do
     alpha := R();
     g := Gcd( f, (Powmod(x-alpha,(p-1)/2,f,x) mod p) - 1 ) mod p:
     deg := degree(g);
     F[deg] := F[deg] + 1;
  od:
```

$$N := 10000$$

```
> convert(F[30..50],list);
```

[0, 0, 2, 6, 7, 6, 18, 18, 44, 76, 100, 162, 244, 311, 418, 487, 578, 685, 735, 765, 744]

```
> data := [seq( i$F[i], i=0..d )]:
> dataplot(data,histogram,discrete,view=[30..70,default],thickness=5)
  ;
```



Binomial $p = \frac{1}{2}$ $n = 100$

$\mu = n \cdot p = 50$

$\sigma = \sqrt{np(1-p)} = \sqrt{25} = 5$

f	whattype(f)	op(0,f)	nops(f)	op(1,f)	op(2,f)
-3	integer	Integer	1	-3	ERROR
$2/3$	fraction	Fraction	2	2	3
3.14	float	Float	2	314	-2
[a,b,b]	list	list	3	a	b
{a,b,b}	set	set	2	a	b
$3x^4y$	`*`	`*`	3	3	x^4
$-y$	`*`	`*`	2	-1	y
x^4	`^`	`^`	2	x	4
$1/x$	`^`	`^`	2	x	-1
$2x^2-3y+\sin t$	`+`	`+`	3	$2x^2$	$-3y$
exp(x)	function	exp	1	x	ERROR
diff(f(x),x)	function	diff	2	f(x)	x
y	symbol	Symbol	1	y	ERROR
a[1,2,3]	indexed	a	3	1	2
a(1,2,3)	function	a	3	1	2

if type(x, symbol) or type(x, indexed) then ....

≡ if type(x, name) then ...

if type(x, integer) or type(x, fraction) or type(x, float) then ....

≡ if type(x, numeric) then ...

type algebraic includes all types above except list, set

```
> unprotect(Diff);
> Diff := proc(f::algebraic,x::name) local u,v,n,y;
     if type(f,numeric) then 0
     elif type(f,name) then if f=x then 1 else 0 fi
     elif op(0,f)=`+` then add( Diff(u,x), u=f );
     elif op(0,f)=`*` then
        u := op(1,f); v := subsop(1=1,f);
        Diff(u,x)*v + Diff(v,x)*u
     elif op(0,f)=`^` and Diff(op(2,f),x)=0 then
        n := op(2,f); u := op(1,f);
        n*Diff(u,x)*u^(n-1)
     elif op(0,f)=exp then
        u := op(1,f); Diff(u,x)*exp(u);
     elif op(0,f)=int and type(op(2,f),name) then
        y := op(2,f); u := op(1,f);
        if x=y then u   # Diff( int(u(x),x), x ) ==> u(x)
        else int( Diff(u,x), y ); # Diff( int(u(x,y),y), x )
        fi
     elif type(f,function) and nops(f)=1 and Diff(op(1,f),x)=0 then 0
     else 'Diff'(f,x)
     fi
  end:
> Diff(2+x+3*x^2,x);
```

$$1+6x$$

```
> g := y*x^2*z-x*y+2*y*z;
```

$$g := y\,x^2\,z - x\,y + 2\,y\,z$$

```
> Diff(g,x);
```

$$2\,x\,z\,y - y$$

```
> Diff(u(x)/v(x)+1/w(x),x);
```

$$\frac{\frac{d}{dx}\,u(x)}{v(x)} - \frac{\left(\frac{d}{dx}\,v(x)\right)u(x)}{v(x)^2} - \frac{\frac{d}{dx}\,w(x)}{w(x)^2}$$

```
> Diff( exp(-2*x)+sin(2*x)+ln(2)*x, x );
```

$$-2\,e^{-2\,x} + \frac{d}{dx}\,\sin(2\,x) + \ln(2)$$

```
> g := int(2*x*f(t),t);
```

$$g := \int 2\,x\,f(t)\,dt$$

```
> Diff(g,t);
```

$$2\,x\,f(t)$$

```
> Diff(g,x);
```

$$\int 2\,f(t)\,dt$$