KaratsubaFFTAgreg

```
def Karatsuba(P,Q,n):
    if n==1:
        return [P[0]*Q[0],0]
    m=n//2
    P0=[P[i] for i in range(m)]
    P1=[P[i+m] for i in range(m)]
    Q0=[Q[i] for i in range(m)]
    Q1=[Q[i+m] for i in range(m)]
    SP=[P0[i]+P1[i] for i in range(m)]
    SQ=[Q0[i]+Q1[i] for i in range(m)]
    A=Karatsuba(P0,Q0,m)
    B=Karatsuba(P1,Q1,m)
    C=Karatsuba(SP,SQ,m)
    R=A+B
    for i in range(n):
        R[i+m]=R[i+m]+C[i]-A[i]-B[i]
    return R
```

```
A=PolynomialRing(ZZ,4,'a')
```

```
v=list(gens(A))
```

```
v
```

```
[a0, a1, a2, a3]
```

```
Karatsuba([v[0],v[1]],[v[2],v[3]],2)
```

```
[a0*a2, a1*a2 + a0*a3, a1*a3, 0]
```

```
len(7.binary())
```

```
3
```

```
a=7
```

```
a.bits()
```

```
[1, 1, 1]
```

```
Qx.<x>=PolynomialRing(QQ)
```

```
P=Qx.random_element()
```

```
P
```

```
x^2 - 3/13*x - 18
```

```
list(P)
```

```
[-18, -3/13, 1]
```

```
def MulK(A,P,Q):
    P=A(P)
    Q=A(Q)
    dP=P.degree()
    dQ=Q.degree()
    d=max(dP,dQ)
    n=2^(len(d.bits()))
    P=list(P)+[0 for i in range(n-dP-1)]
    Q=list(Q)+[0 for i in range(n-dQ-1)]
    R=Karatsuba(P,Q,n)
```

```
    return A(R)
```

```
MulK(Qx,P,P)
```

```
x^4 - 6/13*x^3 - 6075/169*x^2 + 108/13*x + 324
```

```
P^2
```

```
x^4 - 6/13*x^3 - 6075/169*x^2 + 108/13*x + 324
```

```
QQ.random_element(num_bound=100,den_bound=100)
```

```
-18/97
```

```
P=Qx.random_element(degree=(0,100),num_bound=100,den_bound=100)
```

```
Q=Qx.random_element(degree=(0,100),num_bound=100,den_bound=100)
```

```
MulK(Qx,P,Q)-P*Q
```

```
0
```

```
A=PolynomialRing(ZZ,8,'a')
```

```
v=list(gens(A))
```

```
v
```

```
[a0, a1, a2, a3, a4, a5, a6, a7]
```

```
Ax.<x>=PolynomialRing(A)
```

```
P=sum([v[i]*x^i for i in range(4)])
```

```
Q=sum([v[i+4]*x^i for i in range(4)])
```

```
MulK(Ax,P,Q)
```

```
a3*a7*x^6 + (a3*a6 + a2*a7)*x^5 + (a3*a5 + a2*a6 + a1*a7)*x^4 +
(a3*a4 + a2*a5 + a1*a6 + a0*a7)*x^3 + (a2*a4 + a1*a5 + a0*a6)*x^2 +
(a1*a4 + a0*a5)*x + a0*a4
```

```
P*Q
```

```
a3*a7*x^6 + (a3*a6 + a2*a7)*x^5 + (a3*a5 + a2*a6 + a1*a7)*x^4 +
(a3*a4 + a2*a5 + a1*a6 + a0*a7)*x^3 + (a2*a4 + a1*a5 + a0*a6)*x^2 +
(a1*a4 + a0*a5)*x + a0*a4
```

```
MulK(Ax,P,Q)-P*Q
```

```
0
```

```
def FFTrec(n,P,W,k):
    if n==1:
        return P
    m=n//2
    P0=[P[2*i] for i in range(m)]
    P1=[P[1+2*i] for i in range(m)]
    R0=FFTrec(m,P0,W,2*k)
    R1=FFTrec(m,P1,W,2*k)
    R=[0 for i in range(n)]
    for p in range(m):
        R[p]=R0[p]+W[k*p]*R1[p]
        R[p+m]=R0[p]-W[k*p]*R1[p]
    return R
```

```
Qx.<x>=PolynomialRing(QQ)
```

```
P=x^3+x+1
```

```
FFTrec(4,P,[1,I],1)
```

```
x^3 + x + 1
[1, 0]
[1, 1]
[3, 1, -1, 1]
```

```
def FFT(n,P,w):
    W=[1,w]
    for i in range(2,n//2):
        W.append(W[i-1]*w)
    return FFTrec(n,P,W,1)
```

```
FFT(8,P,exp(I*pi/4).n())
```

```
[3,
 1.00000000000000 + 1.41421356237309*I,
 1.00000000000000,
 1.00000000000000 + 1.41421356237309*I,
 -1,
 1.00000000000000 - 1.41421356237309*I,
 1.00000000000000,
 1.00000000000000 - 1.41421356237309*I]
```

```
for i in range(8) :
    print(P(exp(I*pi*i/4).n()))
```

```
3.00000000000000
1.00000000000000 + 1.41421356237309*I
1.00000000000000
1.00000000000000 + 1.41421356237309*I
-1.00000000000000
```

```
1.00000000000000 - 1.41421356237309*I
1.00000000000000
1.00000000000000 - 1.41421356237309*I
```

```
def FFT(n,P,w):
    W=[1,w]
    for i in range(2,n//2):
        W.append(W[i-1]*w)
    print(W)
    return FFTrec(n,P,W,1)
```

```
factor(129)
```

```
3 * 43
```

```
for i in range(10):
    print(i,factor(2^i+1))
```

```
(0, 2)
(1, 3)
(2, 5)
(3, 3^2)
(4, 17)
(5, 3 * 11)
(6, 5 * 13)
(7, 3 * 43)
(8, 257)
(9, 3^3 * 19)
```

```
F=Integers(257)
```

```
F257.multiplicative_generator()
```

```
3
```

```
Fx.<x>=PolynomialRing(F257)
```

```
P=Fx.random_element(degree=(0,15));P
```

```
39*x^15 + 9*x^14 + 6*x^13 + 139*x^12 + 213*x^11 + 59*x^10 + 107*x^9
+ 44*x^8 + 7*x^7 + 210*x^6 + 204*x^5 + 151*x^4 + 22*x^3 + 45*x^2 +
48*x + 155
```

```
FFT(16,P,Fx(3)^16)
```

```
[1, 249, 64, 2, 241, 128, 4, 225]
[2, 250, 65, 3, 242, 129, 5, 226, 0, 9, 194, 256, 17, 130, 254, 33]
```

```
[Fx(3)^i for i in range(16)]
```

```
[1, 3, 9, 27, 81, 243, 215, 131, 136, 151, 196, 74, 222, 152, 199,
83]
```

```
[P(3^(16*i)) for i in range(16)]
```

```
[2, 250, 65, 3, 242, 129, 5, 226, 0, 9, 194, 256, 17, 130, 254, 33]
```

```
P(F(3))
```

```
113
```

```
256/16
```

```
16
```

```
P=x+1
```

```
J=range(8)
```

```
A=[RR(1) for i in range(8)]
```

```
A
```

```
[1.00000000000000,
 1.00000000000000,
 1.00000000000000,
 1.00000000000000,
 1.00000000000000,
 1.00000000000000,
 1.00000000000000,
 1.00000000000000]
```

```
s=IndexedSequence(A,J);s
```

```
Indexed sequence: [1.00000000000000, 1.00000000000000,
1.00000000000000, 1.00000000000000, 1.00000000000000,
1.00000000000000, 1.00000000000000, 1.00000000000000]
    indexed by [0, 1, 2, 3, 4, 5, 6, 7]
```

```
t=s.fft();t
```

```
Indexed sequence: [8.00000000000000, 0.000000000000000,
0.000000000000000, 0.000000000000000, 0.000000000000000,
0.000000000000000, 0.000000000000000, 0.000000000000000]
    indexed by [0, 1, 2, 3, 4, 5, 6, 7]
```

```
lt=t.list();t
```

```
Indexed sequence: [8.00000000000000, 0.000000000000000,
0.000000000000000, 0.000000000000000, 0.000000000000000,
0.000000000000000, 0.000000000000000, 0.000000000000000]
    indexed by [0, 1, 2, 3, 4, 5, 6, 7]
```

```
lt
```

```
[8.00000000000000, 0.000000000000000, 0.000000000000000,
0.000000000000000, 0.000000000000000, 0.000000000000000,
0.000000000000000, 0.000000000000000]
```

```
Qx.<x>=PolynomialRing(QQ)
```

```
P=x^3+x^2+2
```

```
Pl=P.list()
```

```
P.fft()
```

```
Traceback (click to the left of this block for traceback)
...
AttributeError:
'sage.rings.polynomial.polynomial_rational_flint.Polynomial_rational\
_flint' object has no attribute 'fft'
```

```
s=IndexedSequence(Pl,range(4))
```

```
s
```

```
Indexed sequence: [2, 0, 1, 1]
    indexed by [0, 1, 2, 3]
```

```
s.fft()
```

```
Indexed sequence: [4.00000000000000, 1.00000000000000 +
1.00000000000000*I, 2.00000000000000, 1.00000000000000 -
1.00000000000000*I]
    indexed by [0, 1, 2, 3]
```

```
P(-I)
```

```
I + 1
```