CoCoNuT Assignment Three

February 10, 2015

1 More Sage

Integer Rings

The ring $\mathbb{Z}_n$ can be defined using the Integers command. For example:

```
sage: Z7= Integers(7)
sage: Z7
Ring of integers modulo 7
sage: Z7.order()
7
sage: a=Z7(3); b=Z7(4)
sage: a+b
0
```

You can also use `Zmod` to define rings of integers mod n. For example:

```
sage: Z7= Zmod(7)
sage: Z7
Ring of integers modulo 7
```

You can use the method `random_element()` to get a random element from the ring. In the above examples, if you type `Z7.` and then press the Tab key, Sage will return a list of all the methods `Z7` has.

Matrix Rings

The following example defines the ring `MR` containing all 3×3 matrices with entries in $\mathbb{Z}_{51}$.

```
sage: MR = MatrixSpace(Integers(51),3,3)
sage: MR
Full MatrixSpace of 3 by 3 dense matrices over Ring of integers modulo 51
sage: MR.random_element()
[25 21 32]
[25 13 47]
[32 14 41]
```

Polynomial Rings

Example of defining a univariate polynomial ring in Sage.

```
sage: K = Integers(10001)
sage: R.<x> = PolynomialRing(K)
sage: R
Univariate Polynomial Ring in x over Ring of integers modulo 10001
```

Alternatievely you can use:

```
sage: K = Integers(10001)
sage: R.<x> = K[]
sage: R
Univariate Polynomial Ring in x over Ring of integers modulo 10001
```

In the above two examples R defines the ring $\mathbb{Z}_{10001}[x]$, i.e. the ring of polynomials (in the indeterminate x) with coefficients in $\mathbb{Z}_{10001}$.

You can similarly define multivariate polynomial rings, for example:

```
sage: K = Integers(101)
sage: R.<x,y> = K[]
sage: R
Multivariate Polynomial Ring in x, y over Ring of integers modulo 101
```

You can use the method `random_element(n)` to get a random polynomial of degree n from the ring. For example:

```
sage: K = Integers(10001)
sage: R.<x> = K[]
sage: R.random_element(3)
2648*x^3 + 8166*x^2 + 6712*x + 8114
```

Quotients of Polynomial Rings

Examples of defining quotients of polynomial rings $R/p(x)$ for some polynomial ring R and a polynomial $p(x)$, i.e. the ring of polynomials modulo the polynomial $p(x)$. For example, to define $\mathbb{Z}_{11}[x]/(x^2+3x)$

```
sage: Z11=Integers(11)
sage: R.<x>=Z11[]
sage: QR.<y>=R.quotient(x^2+3*x)
sage: QR
Univariate Quotient Polynomial Ring in y over Ring of integers modulo 11 with modulus x^2 + 3*x
sage: QR.order()
121
sage: QR.modulus()
x^2 + 3*x
```

In the above code, one could replace the line `sage: QR.<y>=R.quotient(x^2+3*x)` by `sage: QR=R.quotient(x^2+3*x,'y')` which will result in the same thing.

```
sage: QR.random_element()
6*y + 8
```

2 Assignment Three Questions

1. (a) Using your factoring algorithm `MyFactor` from Sheet 1, write a function `MyPhiFun(`n`)` that computes the Euler totient function (i.e. the phi function) for the integer n.

   Answer:

```
def MyPhiFun(n):
    if n==1:
       return 1
    elif is_prime(n):
       return n-1
    else:
       factors = MyFactor(n)
       phi=n
       for i in range(len(factors)):
           phi = phi * (1 - 1/(factors[i][0]))
       return phi
```

   (b) What does you function output for $n = 42901741984719$?

   Answer:

   28514752980120

2. (a) Write a function `FindNoOfGens` that receives a prime number p and computes the number of generators of the group U(p), i.e. the group of units of the ring of integers modulo p.

   Answer:

```
def FindNoOfGens(p):
    if not is_prime(p) or p==2:
      print "p is not a prime > 2"
```

```
    return -1
  else:
    return euler_phi(p-1)
```

(b) Write your own function that returns the list of the generators of such a group. You are only allowed to call the functions you wrote previously.

Answer:

```
def ListofGens(p):
    if  (not is_prime(p)) or p ==2 :
      print "p is not prime > 2"
      return []
    elif p == 3:
      return [2]
    else:
      gens = []
      factors = MyFactor(p-1)
      for a in range(1,p):
          flag = 0
          for i in range(len(factors)):
             if mod(a,p)^((p-1)//(factors[i][0])) == 1:
                flag = 1
                break
          if flag == 0 : gens.append(a)
      return gens
```

3. Determine which of the following polynomials are irreducible in $\mathbb{Z}_{11}$:

i) $2x^5 + 8x^4 + 3x^3 + 6x^2 + 4x + 1$ ☐
ii) $8x^6 + 3x^5 + 6x^4 + 9x^3 + 5x^2 + 7x + 1$ ✔
iii) $7x^7 + 6x^6 + 2x^5 + 6x^4 + 2x^3 + 10$ ☐

4. In each of the following cases, first find the GCD of $p(x)$ and $q(x)$ and then find the polynomials $a(x)$ and $b(x)$ satisfying $a(x)p(x) + b(x)q(x) = GCD(p(x), q(x))$.

   (a) Take $p(x)$ and $q(x)$ in $\mathbb{Z}_7[x]$ where

$$p(x) = 4x^5 + 3x^4 + x^3 + 6x^2 + 4,$$
$$q(x) = 4x^3 + 5x^2 + x + 4$$

   Answer:

```
GCD=1,
a(x)=3x^2 + 3x + 3,
b(x)=4x^4 + 2x^3 + x^2 + 6x + 6.
```

   (b) Take $p(x)$ and $q(x)$ in $\mathbb{Z}_{13}[x]$ where

$$p(x) = 2x^5 + 10x^4 + 6x^3 + 11x^2 + 10x,$$
$$q(x) = 2x^3 + 2x^2 + 10x + 8.$$

   Answer:

```
GCD=x + 7,
a(x)=7x + 1,
b(x)=6x^3 + 10x^2 + 12x + 9.
```

5. Let $\mathbb{Z}_{17}[x]/p(x)$ be the quotient of a polynomial ring, i.e. the ring of polynomials with coefficients in $\mathbb{Z}_{17}$ modulo the polynomial $p(x)$.

   (a) For each of the following choices of $p(x)$, decide whether or not there exist a polynomial $b(x) \neq 0$ in $\mathbb{Z}_{17}[x]/p(x)$ for which there is no polynomial $a(x) \in \mathbb{Z}_{17}[x]/p(x)$ satisfying $a(x)b(x) = 1 \bmod p(x)$. If in any case your answer is yes, give three different such $b(x)$. You must give the code you used to come up with your answers.

      i. $x^5 + 5x^4 + 7x^3 + 11x^2 + 14x + 11$.

         A. Yes/No :

         B. Examples (if any):

         Answer:

```
None, as p(x) is irreducible
```

C. Code if any):

ii. $x^5 + x^4 + 10x^3 + 4x^2 + 4x + 4$

A. Yes/No : ✔

B. Examples (if any):

Answer:

```
Examples include:
11y^4 + 10y^3 + 14y^2 + 2y + 16
11y^4 + 10y^2 + 14y + 14
2y^3 + 3y^2 + 6y + 3
10y^4 + 5y^3 + 15y^2 + 16y + 11
5y^4 + 10y^3 + 12y^2 + 8y + 9
15y^4 + 7y^3 + 12y^2 + 1
```

C. Code if any):

Answer:

```
Z17=Integers(17)
```

```
R17.<x>=Z17[]
QR17=R17.quotient(x^5 + x^4 + 10*x^3 + 4*x^2 + 4*x + 4)
c=0
b=QR17.random_element()
while (c<5):
  while(b.is_unit()==true):
     b=QR17.random_element()
  print b;
  c = c + 1
  b=QR17.random_element()
```

(b) What can you conclude from such an observation?

Answer:

Such a ring is only a field when p(x) is irreducible.