2014-01-13.sagews

January 13, 2014

Contents

1 Lecture for January 13, 2014

1.1 Startling Observation:

We can test a large number n for primality quickly without factoring n The example below works, but so does any large-ish number.

```
# a 1000-digit random number
set_random_seed(0)
n = ZZ.random_element(0,10^1000)
n
```

```
568967683894693854368083917737833724069101382644691553456920080166342823691147253037686088
451787085232561578538737100443852638310154004892367273326067881574104600550529274820537371
004921197793415509503347182650397980291111953779754682320730090536188068999004114400893442
181426435206304692847647231470090610657379834253138318372794314810574975365398606850996991
746437878408477684774536616037917112978078913565542883282411674952707351475856988081348305
345366324589724443924835794399073364757285797937315828362192002360560879749404954667724675
738763832528731697998411872285449594333406897354946181489871659316182634013306108131221264
914613031868284352865759946371074244432809610255641866684503997179716639845306370103186467
670926407214192520581103346530931133122718454134401211214184185278133134682909938767298956
030254359785260149885128790186990318655557442230426561195163877681704293798877478129090888
241932978670684030394766339980974947771520222651309826454240312653213842916712583485003109
5141004619
```

```
%time n.is_prime()
```

```
False
CPU time: 0.05 s, Wall time: 0.05 s
```

```
n.trial_division()
```

```
5444407
```

1.2 Another Observation:

We can compute the last few digits of certain numbers very quickly without computing all digits of the number.

For example, lets compute the last 10 digits of the largest known prime.

```
%time p = 2^57885161 - 1
```

```
CPU time: 0.01 s, Wall time: 0.02 s
```

```
%time p%(10^10)
```

```
1724285951
CPU time: 0.01 s, Wall time: 0.02 s
```

```
# Now do it the hard way -- should take about 15 seconds.
%time s = str(p)
print s[-10:]
```

```
CPU time: 14.84 s, Wall time: 14.79 s

1724285951
```

Exercise: How could you efficiently compute the first few digits of the number p above very efficiently?

1.2.1 The Ring of Integers Modulo n

Following section 2.1 of the book.

- Define group
- Define abelian group
- Define ring
- Define field
- Definition of congruence and the ring of integers modulo n.
- Prove a few things from section 2.1:
  - gcd(c,n)=1 and ac=bc(mod n) == a=c
  - fact about complete set of residues
  - uniqueness of slution of ax=b (mod n).

```
R = IntegerModRing(20); R
```

```
Ring of integers modulo 20
```

```
list(R)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

```
R(7) + R(18)
```

```
5
```

```
R(3)*R(8)
```

```
4
```