

CoCoALib - Feature #1155

Create a new "prime source" iterator

11 Feb 2018 20:55 - John Abbott

Status:	Closed	Start date:	11 Feb 2018
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	New Function	Estimated time:	22.20 hours
Target version:	CoCoALib-0.99600	Spent time:	21.70 hours
Description			
In CoCoA-4 the old factorizer code had an "iterator" for generating primes in succession. Add such an object to CoCoALib			
Related issues:			
Related to CoCoALib - Feature #1154: SmallFpImpl: new ctor arg to say do-not...		Closed	11 Feb 2018

History

#1 - 11 Feb 2018 21:02 - John Abbott

Add a new class to CoCoALib for generating primes in succession -- for use inside CRT loops.

Currently a CRT loop looks like this:

```
long p;
while (true)
{
    p = NextPrime(p);
    SmallFpImpl ModP(p);
    // computation modulo p
}
```

NextPrime simply increments its arg (in a "clever" way) and then calls IsPrime (or IsSmallPrime), and keeps going until IsPrime produces true.

However it is known that Eratosthenes's Sieve is a quick way of generating a table of primes over a given range, so it may be a good idea to have a PrimeSource object which generates an internal table, then uses that.

So the loop could look like this:

```
PrimeSource PS(1000000); // arg is start value
while (true)
{
    long p = NextPrime(PS);
    SmallFpImpl ModP(p, NoCheck);
    // do computation mod p
}
```

#2 - 11 Feb 2018 21:07 - John Abbott

- Related to Feature #1154: SmallFpImpl: new ctor arg to say do-not-check-that-arg-is-prime added

#3 - 09 Mar 2018 18:08 - John Abbott

- Status changed from New to Resolved

- Assignee set to John Abbott

- % Done changed from 0 to 70

I have added 4 "iterators: **PrimeSeq**, **PrimeSeqForCRT**, **FastMostlyPrimeSeq**, **NoSmallFactorsSeq**.

First trials have been promising.

#4 - 12 Jun 2018 18:35 - John Abbott

- Status changed from Resolved to Feedback

- % Done changed from 70 to 90

Time to move to feedback.

Only remaining doubt: the function NextPrime needs to be reconsidered (what happens when the iterator reaches the end?)

#5 - 03 Aug 2018 16:11 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

- Estimated time set to 22.20 h

SUMMARY

Created 5 new iterators for generating primes (or "almost primes")

Created new type SmallPrime

All seems to work well (except for semantic doubt about NextPrime when it reaches the end)

Closing