CoCoALib - Slug #691

Matrix determinant over ZZ

29 Apr 2015 10:31 - John Abbott

Status:	Closed	**Start date:**	29 Apr 2015
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Improving	**Estimated time:**	1.33 hour
Target version:	CoCoALib-0.99650 November 2019	**Spent time:**	1.30 hour

Description

I tried computing the following:

```
M := Mat(ZZ,[[random(-9,9) | i in 1..500] | j in 1..500]);
D := det(M); // how much time?
```

Which is faster?

- **A** CoCoA-5.1.2 + CoCoALib 0.99536 on 2.4GHz Intel Core 2 duo
- **B** CoCoA-4.5 on 1.07GHz PowerPC G4

Answer: they took about the same time!

Related issues:

Related to CoCoALib - Slug #1110: Determinant of matrix over QQ (whose entrie...	**Closed**	**25 Oct 2017**
Related to CoCoALib - Feature #11: Bareiss algorithm	**Closed**	**20 Oct 2011**
Related to CoCoALib - Feature #1278: Port old "clever" code for matrix determ...	**New**	**03 May 2019**

History

#1 - 29 Apr 2015 10:32 - John Abbott

What?!? My dad's ancient iBook G4 is just as fast as my much newer MacBook Pro???

Something must be seriously wrong!

#2 - 29 Apr 2015 11:22 - John Abbott

I've just tried a 1000x1000 matrix: the old G4 took about 550s, my newer MacBook Pro took 1400s. Embarrassing!

NOTE CoCoA-4.7.6 on MacBook Pro took about 220s

#3 - 29 Apr 2015 11:33 - John Abbott

Just as a speed comparison reference I computed 3^(3^17) on both machines:

- old G4 took about 50s (on battery power)
- new MacBook Pro took 2.8s (on mains)

Of course the underlying version of GMP is not the same, but I suspect that they are fairly similar for power/product of integers. So the newer machine is about 20 times faster than the old one.

#4 - 29 Apr 2015 15:46 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

Hint: DenseMatrix.C:462 the dispatch function DenseMatImpl::myDet does not handle matrices over ZZ specially -- they are just passed to a Bareiss impl.

#5 - 25 Oct 2017 13:29 - John Abbott

- Related to Slug #1110: Determinant of matrix over QQ (whose entries are actually integers) added

#6 - 26 Jun 2018 14:38 - John Abbott

CoCoALib currently uses DetByCRT rather than the "clever" algorithms used in CoCoA-4.7. This issue should be about porting that old code to CoCoALib.

Anyway, the times on my Linux laptop are now about 75s for a 1000x1000 matrix, and about 4.5s for a 500x500 matrix.

#7 - 03 May 2019 11:35 - John Abbott

- Status changed from In Progress to Closed

- Assignee set to John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99700

- % Done changed from 10 to 100

- Estimated time set to 1.33 h

I think this was resolved some time ago when I arranged for the code to recognise the case that the entries are all integers. Anyway, this improved code is acceptably fast (for the time being).

Closing, but will create a new issue to port the old "clever" CoCoA-4 code to CoCoALib.

#8 - 03 May 2019 11:39 - John Abbott

- Related to Feature #11: Bareiss algorithm added

#9 - 03 May 2019 11:42 - John Abbott

- Related to Feature #1278: Port old "clever" code for matrix determinant over ZZ to CoCoALib added

#10 - 10 Oct 2019 18:46 - Anna Maria Bigatti

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99650 November 2019