CoCoALib - Slug #1517

RandomLinearForm

23 Oct 2020 16:21 - John Abbott

Status:	Closed	**Start date:**	23 Oct 2020
Priority:	Low	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Improving	**Estimated time:**	0.90 hour
Target version:	CoCoALib-0.99800	**Spent time:**	0.90 hour

Description

The profiler tells me that RandomLinearForm spends most (almost all!) of its time in operator+=.

If the indets are ordered nicely (how can we know this?) then it may be better to use PushFront?

Anyway, RandomLinearForm is disappointingly slow in the test example from issue #1514 (with 1000 indets)

Related issues:

Related to CoCoALib - Feature #1169: New function: RandomLinearForm (CoCoALib)	**Closed**	**19 Mar 2018**
Related to CoCoALib - Bug #1208: New function: Threadsafe RandomLinearForm (...	**New**	**02 Aug 2018**

History

#1 - 23 Oct 2020 16:21 - John Abbott

- Related to Feature #1169: New function: RandomLinearForm (CoCoALib) added

#2 - 23 Oct 2020 16:21 - John Abbott

- Related to Bug #1208: New function: Threadsafe RandomLinearForm (CoCoALib) added

#3 - 23 Oct 2020 16:22 - John Abbott

- Priority changed from Normal to Low

It could be that the problem is simply copying lots of PPs (each occupying 4000 bytes).
Could the memory manager be a bottleneck?

It would be nice to make it faster...

Source code: SparsePolyOps-RingElem.C around line 148

STRANGE operator+= seems to be calling CoCoA::RingDistrMPolyInlFpPPImpl::myAdd which would copy the whole poly, right?

#4 - 29 Oct 2020 14:43 - John Abbott

- Description updated

- Status changed from New to In Progress

- % Done changed from 0 to 10

The empirical complexity appears to be quadratic. Here is my test:

```
nvars := 40000;
use P ::= ZZ/(29641)[x[1..nvars]];
t0 := CpuTime();
```

```
l := RandomLinearForm(P);
TimeFrom(t0);
```

With nvars=20000 the time was about 1.15s; with nvars=40000 the time was about 4.4s.

With our dense repr for ordvs, the memory consumption is also essentially quadratic.

It might be faster with PPMonoidSparse... I wonder what state that code is in?

#5 - 04 Nov 2020 13:35 - John Abbott

- % Done changed from 10 to 50

I presume the timings in comment 4 were with unsigned short as SmallExponent_t (otherwise CoCoA has suddenly become slower by a factor of 2).

RandomLinearForm is definitely faster in CoCoALib, and the time taken is super-linear in the number of indets; but I cannot go far, because with 50000 indets the process dies spontaneously after reaching about 16Gb RAM -- why???

Anyway, since the memory requirement is quadratic there is no hope to make it much faster... perhaps it is not that important anyway?

#6 - 04 Feb 2022 21:31 - John Abbott

- Status changed from In Progress to Closed

- Assignee set to John Abbott

- % Done changed from 50 to 100

- Estimated time set to 0.90 h

The code works. If there is a real use-case where the low speed is a problem then we can create a new issue.
I think we can just close this... I doubt it is that important (and it just clutters up the list of open issues).