

CoCoALib - Slug #722

valuation slow for large inputs

31 May 2015 11:56 - John Abbott

Status:	Closed	Start date:	31 May 2015
Priority:	Low	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Improving	Estimated time:	2.00 hours
Target version:	CoCoALib-0.99800	Spent time:	1.95 hour
Description			
I tried valuation(5,factorial(1000000)) and it took too long. Make valuation faster if the prime is small (and the number big).			
Related issues:			
Related to CoCoA-5 - Design #490: Duplicate fns: valuation and FactorMultipli...		Closed	21 Mar 2014

History

#1 - 31 May 2015 11:58 - John Abbott

- Estimated time set to 2.00 h

Source code is in NumTheory.C.

It is just a very simple loop. Presumably I could test divisibility by a small power of the prime. Might it be useful to compute gcd? Perhaps too costly to get the quotient?

#2 - 01 Jun 2015 10:39 - John Abbott

Probably QuoRem is a more sensible choice than gcd.

#3 - 01 Jun 2015 22:31 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 30

I have implemented a version which divides repeatedly by the largest power of the prime which fits into unsigned long. This now takes about 45s for valuation(2,N) and 26s for valuation(5,N) where N=factorial(1000000).

Maybe a recursive "repeated squaring" version would be neater and faster?

#4 - 29 Jun 2015 14:02 - John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99540 Feb 2016

#5 - 30 Jul 2015 16:16 - John Abbott

- Assignee set to John Abbott

In issue [#490](#) I decided to make the definition of FactorMultiplicity wider: the "base" need not be prime (just an integer greater than 1).

I'm accepting the current impl for the imminent release of C5.1.2 (tomorrow?), but will leave this issue open, so that someone will look at making a proper recursive impl.

#6 - 30 Jul 2015 17:16 - John Abbott

Extra task: extend defn of FactorMultiplicity to RingElem (belonging to a TrueGCDDomain).

#7 - 01 Feb 2016 14:26 - John Abbott

- Target version changed from CoCoALib-0.99540 Feb 2016 to CoCoALib-1.0

#8 - 27 Jan 2020 17:35 - John Abbott

The timings on my current machine (Fujitsu) are:

FactorMultiplicity(2, N) about 16s

FactorMultiplicity(5, N) about 9.7s

NOTE: source code is now in NumTheory-factor.C

#9 - 07 Oct 2020 22:11 - John Abbott

- Status changed from In Progress to Closed

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99800

- % Done changed from 30 to 100

The current impl is "good enough" for the moment. If a specific application comes up where more speed is needed then we can reconsider the impl. I do not see any point in investing time now to make a rather more complicated version (which is perhaps a bit faster in certain rather artificial tests).