

CoCoALib - Bug #895

NumDigits: sometimes gives wrong answer

24 Jun 2016 16:22 - John Abbott

Status:	Closed	Start date:	24 Jun 2016
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Improving	Estimated time:	1.51 hour
Target version:	CoCoALib-0.99550 spring 2017	Spent time:	1.45 hour
Description			
NumDigits is a direct call to the CoCoALib fn called NumDigits which is a direct call to the GMP fn mpz_sizeinbase; this last fn returns either the correct result or the correct result PLUS ONE.			
This ambiguity is inappropriate for a "high-level function" exposed to CoCoA-5 users.			
Fix the situation!			

History

#1 - 24 Jun 2016 16:26 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

The GMP function is guaranteed to be fast (very fast!).

The correct definition is $\text{NumDigits}(N, \text{base}) = 1 + \text{FloorLogBase}(N, \text{base})$.

Probably the only version which would be useful is base 10, so we could define NumDigits10 instead of NumDigits.

The current CoCoALib fn NumDigits is used for conversion to strings in BigInt.C and BigRat.C. There are two possible solutions: replace the call by $1 + \text{FloorLogBase}(N, \text{base})$, or else rename the function (perhaps NumDigits_GMP?).

Comments? Opinions? Ideas?

#2 - 24 Jun 2016 16:53 - John Abbott

Which variants of NumDigits should exist and what should they be called?

- base 10 possible name NumDigits10
- base 2 possible name NumDigits2
- any base possible name NumDigitsBase or simply NumDigits?

My suggestions for these names were inspired by the names of FloorLogXXX which are closely related functions.

Which (if any!) of these should be made visible in CoCoA-5?

Comments? Opinions? Ideas?

#3 - 25 Jun 2016 11:40 - John Abbott

I have just tried a quick experiment comparing speeds: let $N := 10^{10^9}$;

- NumDigits(N,10) takes less than 0.001s -- it uses GMP's function directly
- FloorLog10(N) takes about 40s (on my old MacBook).

I'm not sure how useful the extreme speed of the GMP function is. As a comparison I computed N^2 , and that took about 80s (but it did run into swap problems).

In any case, it seems quite unlikely to me that the extreme speed of the GMP function could be useful to someone programming in CoCoA-5.

NOTE the FloorLog functions are reasonably quick provided the true logarithm is not too close to an integer; for instance FloorLog10(2^*N) takes less than 0.001s (assuming 2^*N was precalculated).

CURIOSITY my old MacBook took about 1.6s to compute 2^*N , but about 2.3s to compute $N+N$

#4 - 02 Aug 2016 11:26 - John Abbott

- Status changed from *In Progress* to *Closed*
- Assignee set to *John Abbott*
- % Done changed from 10 to 100

After discussing with Anna, we have decided to rename NumDigits to SizeInBase which mirrors the GMP fn name. The manual already points out that the value returned may be too large by 1.

The function will not be exported to CoCoA-5 as it is low level, and may easily be imitated (without the risk of producing a wrong answer) by calling FloorLogBase. The speed penalty is unlikely to be important in CoCoA-5 code.

#5 - 02 Aug 2016 11:26 - John Abbott

- Description updated

#6 - 28 Apr 2017 09:30 - Anna Maria Bigatti

- Estimated time set to 1.51 h