

CoCoALib - Bug #1575

DynamicBitset missing arg checks

03 Feb 2021 18:20 - John Abbott

Status:	New	Start date:	03 Feb 2021
Priority:	Normal	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	0%
Category:	Safety	Estimated time:	0.00 hour
Target version:	CoCoALib-0.99880	Spent time:	0.30 hour
Description I have just looked at the code for DynamicBitset (with the intention of putting in noexcept), and noticed that several functions do not check the validity of their args (not even via CoCoA_ASSERT). It would be good to fix this soon!			
Related issues: Related to CoCoALib - Feature #1419: DynamicBitset::lamAll1s New 17 Feb 2020			

History

#1 - 08 Feb 2021 12:13 - John Abbott

- Related to Feature #1419: DynamicBitset::lamAll1s added

#2 - 08 Feb 2021 12:13 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

#3 - 16 Feb 2021 13:43 - John Abbott

My usual approach regarding errors and assertions is:

- if the fn is the main public fn for that operation (and documented) then it should produce errors when bad args are passed
- otherwise check using assertions.

With this approach, errors always have a run-time cost, while assertions cost nothing unless debugging was activated (and everything recompiled). I can see an argument for using assertions instead of errors for short, fast, low-level functions because the overhead of always checking may be annoyingly high.

This last comment does suggest that indexing functions could/should perhaps use assertions instead of errors. This could be a problem for a careless CoCoA-5 user, because a bad index might cause a crash (or worse). Maybe it would be better to offer two such functions: one which does check its args, and one which does not -- and the latter should have a name indicating that checking has been disabled (but it probably should still use CoCoA_ASSERT).

#4 - 03 Nov 2021 16:50 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

#5 - 16 Feb 2024 09:46 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99880

There are a few functions whose name has a suffix like **_NoArgChecks**. OK, a quick grep produces just one such public function InvModNoArgCheck; but there are also some "hidden functions" (e.g. in anonymous

namespaces)