

**NAME**

Algorithm::Diff::XS - Algorithm::Diff with XS core loop

**SYNOPSIS**

```
# Drop-in replacement to Algorithm::Diff, but "compact_diff"
# and C<LCSidx> will run much faster for large data sets.
use Algorithm::Diff::XS qw( compact_diff LCSidx );
```

**DESCRIPTION**

This module is a simple re-packaging of Joe Schaefer's excellent but not very well-known Algorithm::LCS with a drop-in interface that simply re-uses the installed version of the [Algorithm::Diff](#) module.

Note that only the LCSidx function is optimized in XS at the moment, which means only compact\_diff will get significantly faster for large data sets, while diff and sdiff will run in identical speed as [Algorithm::Diff](#)

**BENCHMARK**

```
Rate Algorithm::Diff Algorithm::Diff::XS
Algorithm::Diff 14.7/s -- -98%
Algorithm::Diff::XS 806/s 5402% --
```

The benchmarking script is as below:

```
my @data = ([qw/a b d/ x 50], [qw/b a d c/ x 50]);
cmpthese( 500, {
  'Algorithm::Diff' => sub {
    Algorithm::Diff::compact_diff(@data)
  },
  'Algorithm::Diff::XS' => sub {
    Algorithm::Diff::XS::compact_diff(@data)
  },
});
```

**SEE ALSO**

[Algorithm::Diff](#), [Algorithm::LCS](#).

**AUTHORS**

Audrey Tang <cpan@audreyt.org>

**COPYRIGHT**

Copyright 2008 by Audrey Tang <cpan@audreyt.org>.

Contains derived code copyrighted 2003 by Joe Schaefer, <joe+cpan@sunstarsys.com>.

This library is free software; you can redistribute it and/or modify it under the same terms as Perl itself.