benchmark results

for elm-rope

List

all	runs / second	goodness of fit
recursive	149,782	98.81%
not (any isBad) with let isBad = not ...	56,596	98.36%
not (any (not ...))	55,340	96.6%
not (any (not << ...))	30,792	98.14%

Rope

fold -l vs -r	runs / second	goodness of fit
foldl	5,244	98.29%
foldr	4,584	98.73%

toList	runs / second	goodness of fit
with foldr	4,092	98.9%
with foldr preserving last list	4,015	99.03%
with foldl |> reverse	2,826	98.78%

length	runs / second	goodness of fit
with foldl	5,433	99.12%
with foldr	4,606	99.25%
nested	4,019	99.46%

all	runs / second	goodness of fit
nested	260,560	98.4%
not (any (not << ...))	249,821	98.98%

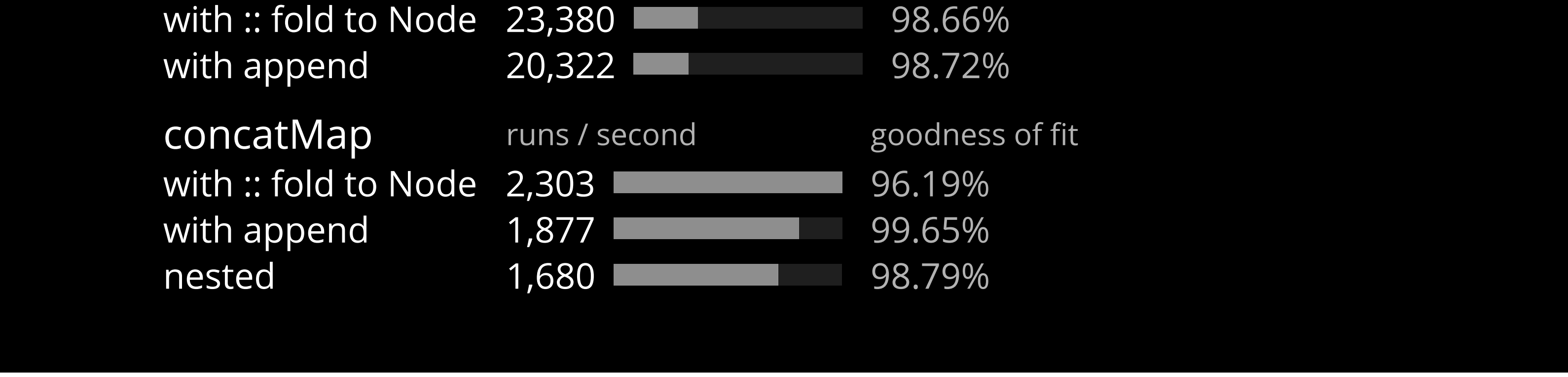
sum (also applies to product)	runs / second	goodness of fit
with foldl	5,173	98.98%
with foldr	4,415	98.84%
nested	4,385	98.91%

minimum (also applies to maximum)	runs / second	goodness of fit
with nested fold	3,305	99.14%
with filterMap	3,104	99.01%
with foldr	2,772	98.36%
with foldl	2,635	98.6%

map	runs / second	goodness of fit
to flat with foldr	3,415	98.86%
to flat with foldl |> reverse	2,526	97.72%
keep nested	2,257	99.12%

indexedMap	runs / second	goodness of fit
with foldl to end	2,202	98.52%
with foldl |> reverse	2,164	98.6%

concat	runs / second	goodness of fit
nested	81,430	99.11%



with :: fold to Node	23,380	98.66%
with append	20,322	98.72%

concatMap	runs / second	goodness of fit
with :: fold to Node	2,303	96.19%
with append	1,877	99.65%
nested	1,680	98.79%