benchmark results

for elm-rope

List

all	runs / second	goodness of fit
recursive	453,228	98.95%
not (any isBad) with let isBad = not ...	374,643	98.81%
not (any (not ...))	373,181	98.94%
not (any (not << ...))	44,256	98.92%

Rope

fold -l vs -r	runs / second	goodness of fit
foldr	10,535	98.86%
foldl	10,303	99.08%

toList	runs / second	goodness of fit
with foldr preserving last list	9,692	98.86%
with foldr	9,670	98.84%
with foldl |> reverse	8,090	99%

length	runs / second	goodness of fit
with foldl	11,093	98.83%
with foldr	11,058	98.92%
nested	10,276	98.64%

all	runs / second	goodness of fit
not (any (not << ...))	541,085	98.76%
nested	497,326	98.8%

sum (also applies to product)	runs / second	goodness of fit
nested	17,987	98.73%
with foldr	10,187	98.39%
with foldl	10,029	98.63%

minimum (also applies to maximum)	runs / second	goodness of fit
with nested fold	9,376	98.97%
with filterMap	8,171	98.65%
with foldr	5,507	98.88%
with foldl	5,431	98.68%

map	runs / second	goodness of fit
to flat with foldr	6,735	98.97%
to flat with foldl |> reverse	5,971	98.81%
keep nested	4,278	99.04%

indexedMap	runs / second	goodness of fit
with foldl |> reverse	4,702	99.06%
with foldl to end	3,569	98.89%

concat	runs / second	goodness of fit

nested	151,466	98.91%
with append	40,740	99%
with :: fold to Node	37,168	98.91%

concatMap	runs / second	goodness of fit
with :: fold to Node	5,684	99.17%
nested	3,694	99.27%
with append	3,559	98.94%