benchmark results

for elm-rope

List

all

	runs / second		goodness of fit
recursive	271,028		99.28%
not (any isBad) with let isBad = not …	236,294		99.35%
not (any (not …))	234,634		99.51%
not (any (not << …))	30,058		99.56%
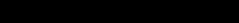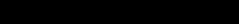
Rope
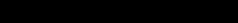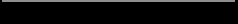
fold -l vs -r

	runs / second	goodness of fit
foldl	6,879	99.81%
foldr	6,097	99.35%

toList

	runs / second	goodness of fit
with foldr preserving last list	5,938	99.39%
with foldr	5,884	99.42%
with foldl |> reverse	5,542	99.41%

all
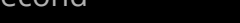
	runs / second	goodness of fit
not (any (not …))	370,419	99.31%
nested	293,633	99.4%

sum (also applies to product)

	runs / second	goodness of fit
nested	12,442	99.43%
with foldl	6,963	99.66%
with foldr	6,194	99.47%

minimum (also applies to maximum)

	runs / second	goodness of fit
with nested fold	6,497	99.36%
with filterMap	5,713	99.55%
with foldl	3,643	99.44%
with foldr	3,612	99.45%

reverse

	runs / second	goodness of fit
keep nested	9,431	99.46%
foldl to flat	6,676	99.52%
foldr to end to flat	2,851	99.67%

filter

	runs / second	goodness of fit
foldr to flat	4,919	99.44%
foldl |> reverse to flat	4,566	99.42%
foldl to end to flat	3,775	99.35%
keep nested	3,087	99.55%

filterMap

	runs / second	goodness of fit
foldr to flat	4,475	99.53%
foldl |> reverse to flat	4,136	99.51%
foldl to end to flat	3,513	99.45%

keep nested   1,508 ▬▬▬ 99.56%

map

	runs / second	goodness of fit
foldr to flat	4,362	99.44%
foldl |> reverse to flat	4,251	99.36%
keep nested	2,844	99.26%

indexedMap

	runs / second	goodness of fit
with foldl |> reverse	3,236	98.92%
with foldl to end	2,464	99.29%

concat

	runs / second	goodness of fit
nested	92,548	99.35%
with :: fold to Node	28,748	99.53%
with append	27,074	99.27%

concatMap

	runs / second	goodness of fit
with :: fold to Node	3,567	98.69%
with append	3,011	98.49%
nested	2,368	99.58%

length

	runs / second	goodness of fit
with foldl	7,511	99.57%
nested	7,158	99.64%
with foldr	6,581	99.56%

isEmpty

	runs / second	goodness of fit
nested	53,374	99.54%
with foldr	15,760	99.5%
with foldl	14,807	99.67%