







benchmark results




for elm-rope


List


all	runs / second	goodness of fit
recursive	185,572 	98.45%
not (any isBad) with let isBad = not ...	95,765 	98.7%
not (any (not ...))	92,889 	99.02%
not (any (not << ...))	45,253 	99.15%

Rope





fold -l vs -r	runs / second	goodness of fit
foldl	9,864 	98.76%
foldr	8,089 	99.04%

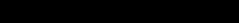
toList	runs / second	goodness of fit
with foldr	7,792 	98.24%
with foldr preserving last list	7,621 	98.58%
with foldl > reverse	4,804 	98.99%

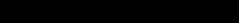
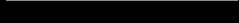
length	runs / second	goodness of fit
nested	11,385 	98.9%
with foldl	9,553 	98.82%
with foldr	8,459 	98.79%

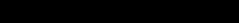
all	runs / second	goodness of fit
nested	442,826 	98.38%
not (any (not << ...))	376,750 	98.67%






sum (also applies to product)	runs / second	goodness of fit
with foldl	9,712 	98.85%
nested	9,328 	98.24%
with foldr	8,436 	99.31%

minimum (also applies to maximum)	runs / second	goodness of fit
with nested fold	5,749 	99.09%
with filterMap	5,074 	98.98%
with foldr	4,818 	98.65%
with foldl	4,813 	99.21%

map	runs / second	goodness of fit
keep nested	6,384 	98.83%
to flat with foldr	6,039 	98.93%
to flat with foldl > reverse	4,464 	98.42%

indexedMap	runs / second	goodness of fit
with foldl > reverse	3,673 	99.11%
with foldl to end	3,362 	98.73%

concat	runs / second	goodness of fit
nested	224,262 	99.14%

with :: fold to Node	43,318		99.01%
with append	40,423		98.86%
concatMap			
	runs / second		goodness of fit
with :: fold to Node	4,238		99.24%
nested	3,580		98.62%
with append	2,839		98.02%