







# benchmark results


## for elm-rope



### List

all	runs / second	goodness of fit
recursive	458,478 	98.35%
not (any (not ...))	424,776 	98.66%
not (any isBad) with let isBad = not ...	422,809 	98.9%
not (any (not << ...))	46,507 	98.72%

### Rope





fold -l vs -r	runs / second	goodness of fit
foldl	15,281 	98.91%
foldr	12,493 	98.65%

toList	runs / second	goodness of fit
with foldr	11,939 	98.85%
with foldr preserving last list	11,674 	99.01%
with foldl  > reverse	11,059 	98.68%



length	runs / second	goodness of fit
nested	40,253 	98.5%
with foldl	16,544 	98.48%
with foldr	13,583 	98.72%


all	runs / second	goodness of fit
nested	714,197 	98.61%
not (any (not << ...))	638,388 	98.18%






sum (also applies to product)	runs / second	goodness of fit
nested	30,420 	98.67%
with foldl	15,457 	98.53%
with foldr	12,616 	98.65%

minimum (also applies to maximum)	runs / second	goodness of fit
with nested fold	15,591 	97.86%
with filterMap	9,231 	98.43%
with foldr	7,822 	98.65%
with foldl	7,740 	98.6%

map	runs / second	goodness of fit
keep nested	13,303 	98.5%
to flat with foldl  > reverse	11,125 	98.35%
to flat with foldr	11,046 	98.94%

indexedMap	runs / second	goodness of fit
with foldl  > reverse	7,301 	98.49%
with foldl to end	4,954 	98.53%

concat	runs / second	goodness of fit
nested	408,940 	98.6%

with append	75,647		98.1%
with :: fold to Node	60,163		98.57%
concatMap	runs / second		goodness of fit
nested	10,453		98.5%
with :: fold to Node	7,610		98.89%
with append	6,861		98.85%