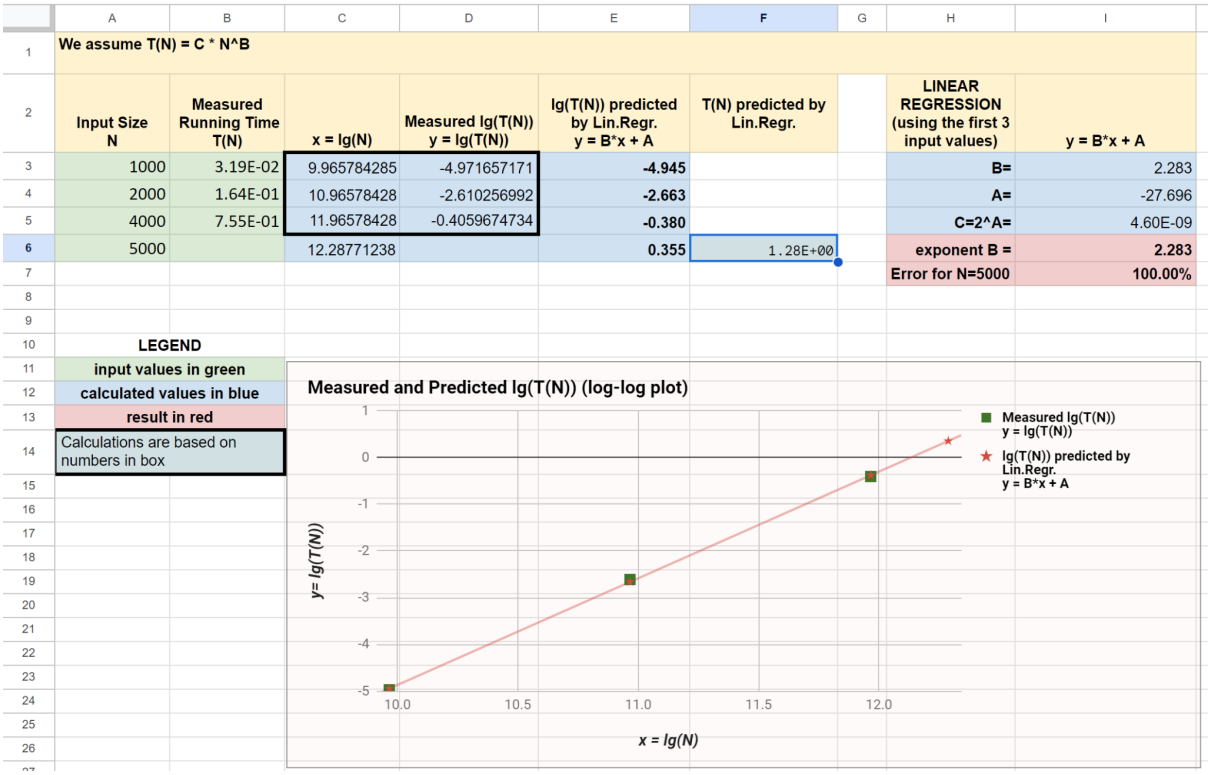
countCollinearFast performance

This function is almost identical to countCollinear but it uses a sorting algorithm and binary searching to speed up the innermost process. In order to test this function accurately I used a loop to run each set of inputs 100 times and calculate the mean average of the samples. This was significantly more time consuming but I did it anyway as I felt it was better experimental practice. The results are displayed in the table below.

Input size	Execution time
1000	0.03187 seconds
2000	0.16377 seconds
4000	0.75473 seconds

After fitting these measurements to a log-log plot with a linear regression, the estimated B value comes to 2.283.



We assume T(N) = C * N^B

Input Size N	Measured Running Time T(N)	x = lg(N)	Measured lg(T(N)) y = lg(T(N))	lg(T(N)) predicted by Lin.Regr. y = B*x + A	T(N) predicted by Lin.Regr.
1000	3.19E-02	9.965784285	-4.971657171	-4.945	
2000	1.64E-01	10.96578428	-2.610256992	-2.663	
4000	7.55E-01	11.96578428	-0.4059674734	-0.380	
5000		12.28771238		0.355	1.28E+00

LINEAR REGRESSION (using the first 3 input values)	y = B*x + A
B=	2.283
A=	-27.696
C=2^A=	4.60E-09
exponent B =	2.283
Error for N=5000	100.00%

LEGEND
input values in green
calculated values in blue
result in red
Calculations are based on numbers in box

Based on this linear regression model, an input size of 5000 would take 1.28 seconds to execute. After running my test, the average execution time was 1.14862 seconds. This is an error of 10.2% which is a little larger than I would like, but not that bad.