

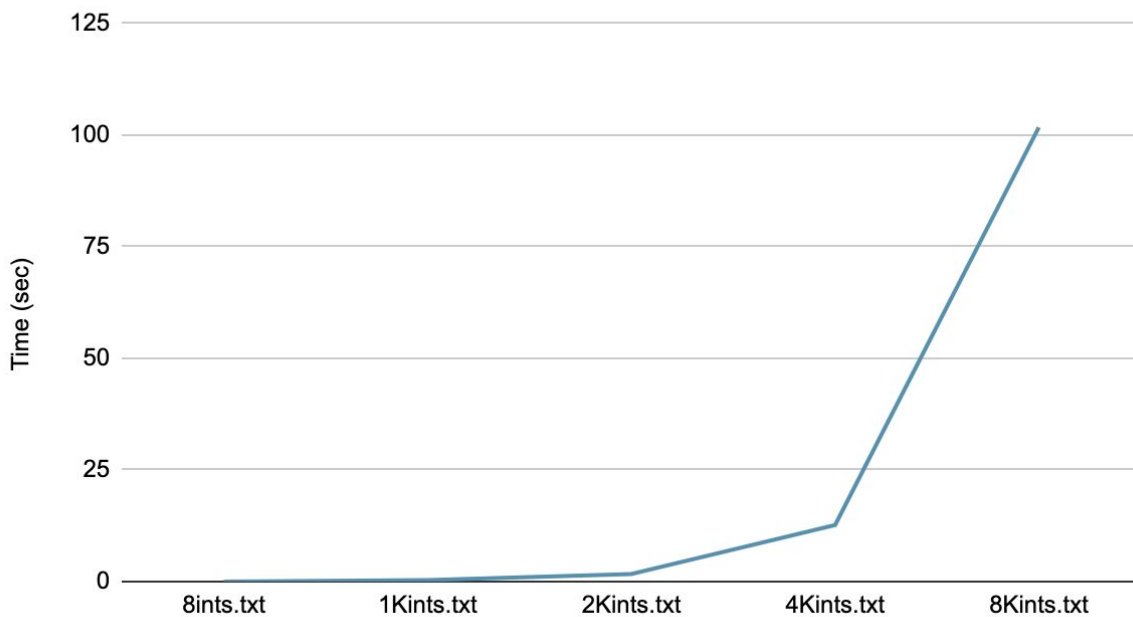
Algorithms - Practical 1

Record your results of your timing experiments in the table below:

Algorithm	Input	Time (s)	Number of Triplets?
ThreeSumA	8ints.txt	0.0	4
	1Kints.txt	0.294	70
	2Kints.txt	1.671	528
	4Kints.txt	12.655	4039
	8Kints.txt	101.713	32074
	16Kints.txt	796.498	255181

Graph for ThreeSumA:

ThreeSumA

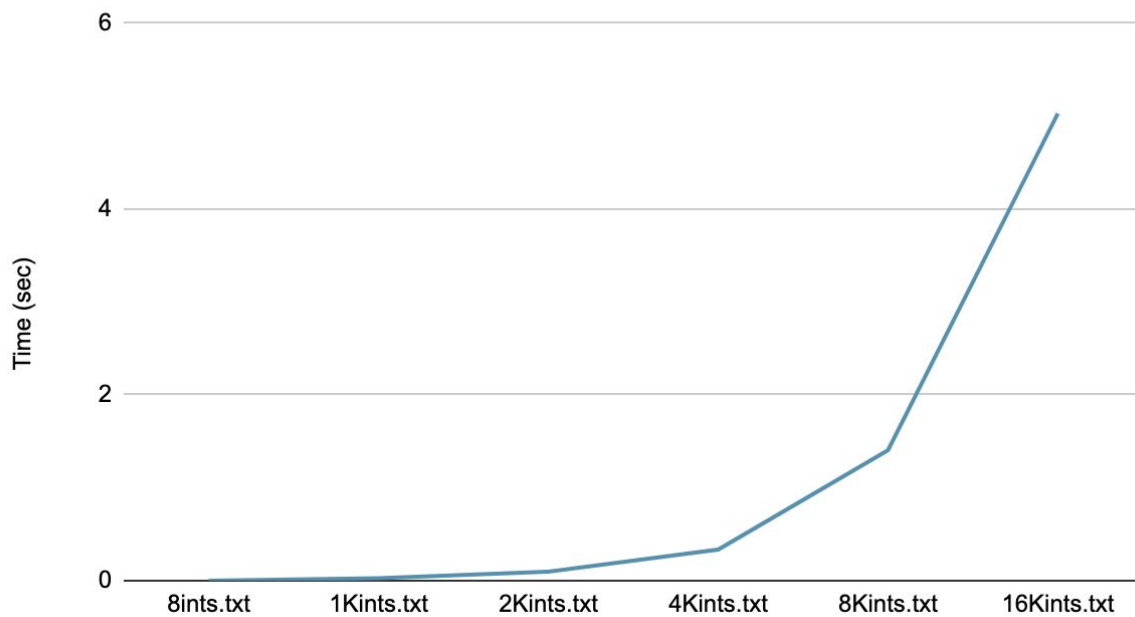


Record your results of your timing experiments in the table below:

Algorithm	Input	Time (s)	Number of Triplets?
ThreeSumB	8ints.txt	0.0	4
	1Kints.txt	0.025	70
	2Kints.txt	0.095	70
	4Kints.txt	0.334	528
	8Kints.txt	1.404	4039
	16Kints.txt	5.03	255181

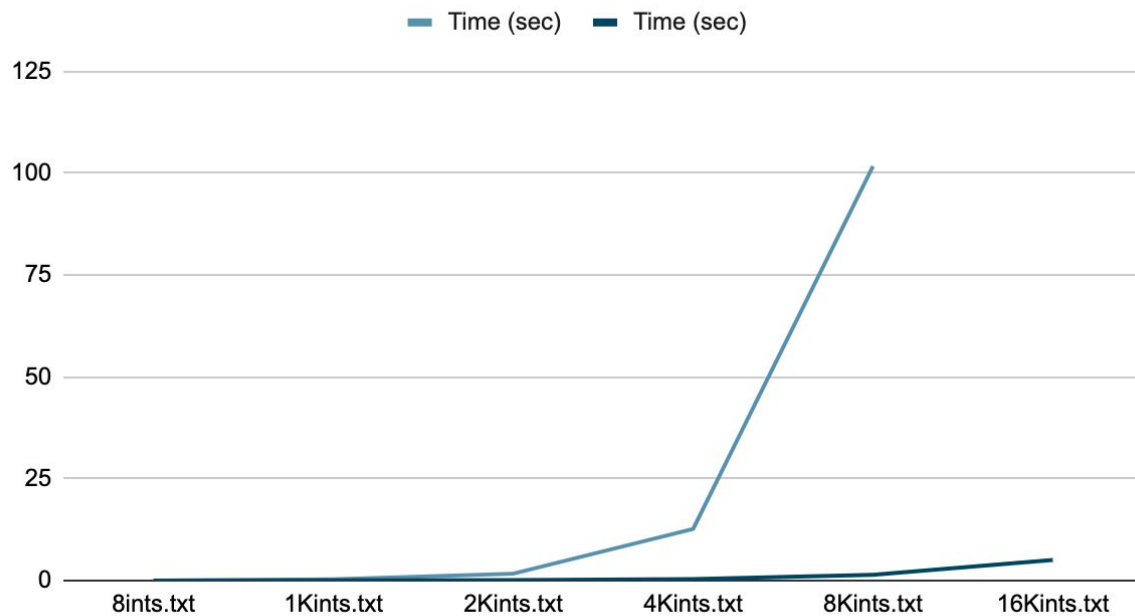
Graph for ThreeSumB:

ThreeSumB



Two graphs together:

ThreeSumA and ThreeSumB



Questions:

1: Which algorithm performs better?

The second algorithm ThreeSumB performs better as less time elapses for the same result as ThreeSumA. The first algorithm took so long that my IDE could not get a result for it. The second algorithm ThreeSumB took 5.03 seconds to calculate the triplets of 8,000 integers that equal 0 whereas the first algorithm ThreeSumA took 101.7 seconds to achieve the same result.

2: Why do you think this is the case?

From the examination of the first algorithm the program must cycle through the integers n^3 times n being the number of integers in the data file. Although this achieves the results required it is very inefficient.

ON the other hand the second algorithm ThreeSumB must only cycle through the total number of integers n , $(n^2 - 1)$ times which is far more efficient than the previous algorithm, especially when the number of integers in the data set increases.