

В Таблице 1 приведено сравнение алгоритмов умножения для структур хранения FlatMatrix и CRSMatrix при одинаковых входных данных (функция main()). Size – порядок матрицы, Frequency – отношение числа ненулевых элементов к числу Size * Size, т. е. процент ненулевых элементов матрицы.

	Size					
	100	200	300	400		
FlatMatrix	78.3052	575.392	2335.13	5187.23	5%	Frequency
	105.511	638.888	2075.52	5045.68	10%	
	64.1426	576.976	2008.71	5082.44	50%	
	111.419	640.515	2179.75	5199.42	100%	
CRSMatrix	5.7228	66.949	218.292	496.032	5%	Frequency
	14.4335	136.976	375.304	994.011	10%	
	64.8481	571.036	1946.98	5102.13	50%	
	164.448	1192.77	3408.01	7943.31	100%	

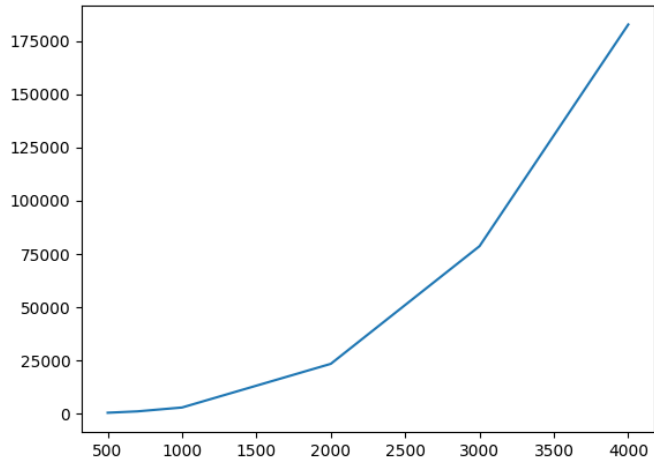
В Таблице 2 приведено среднее время работы алгоритма умножения для структуры хранения CRSMatrix по результатам 10 проходов алгоритма.

		Size		
		10	100	1000
Frequency	1%	0.1233	1.83204	1369.94
	5%	0.10257	9.35429	7286.87
	10%	0.10271	19.7329	13825.3
	20%	0.15788	34.935	27383.7
	50%	0.38107	83.211	69212.6

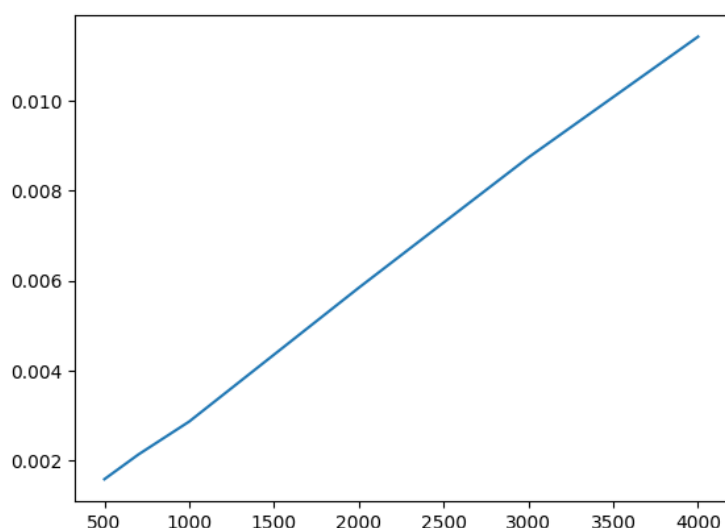
В Таблице 3 приведено среднее время работы алгоритма умножения для структуры хранения CRSMatrix для Frequency = 1

Size	500	700	1000	2000	3000	4000
Time, ms	397.185	1047.2	2871.02	23386.9	78695.5	182874

График зависимости времени от размера матриц:



Полученный график похож на параболу, тогда предположим, что сложность алгоритма умножения в CRSMatrix есть $O(n^2)$. Построим еще один график зависимости, где элементу Size[i] поставим в соответствие элемент $\text{Time}[i] / (\text{Size}[i] ** 2)$:



Видим линейную функцию малого роста, при изменении аргумента на $\Delta \text{size} = 3500$, значение функции изменяется всего лишь на $\Delta \text{time} = 0.08$, что означает, что предположение о сложности $O(n^2)$ условно справедливо.