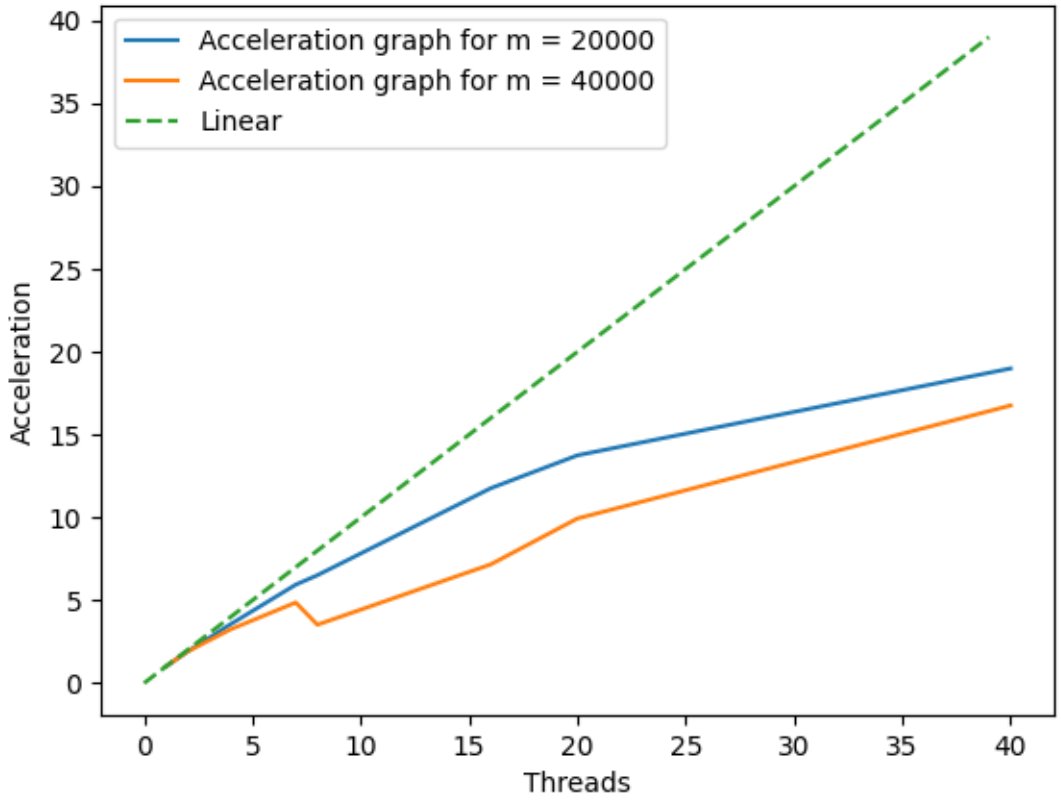
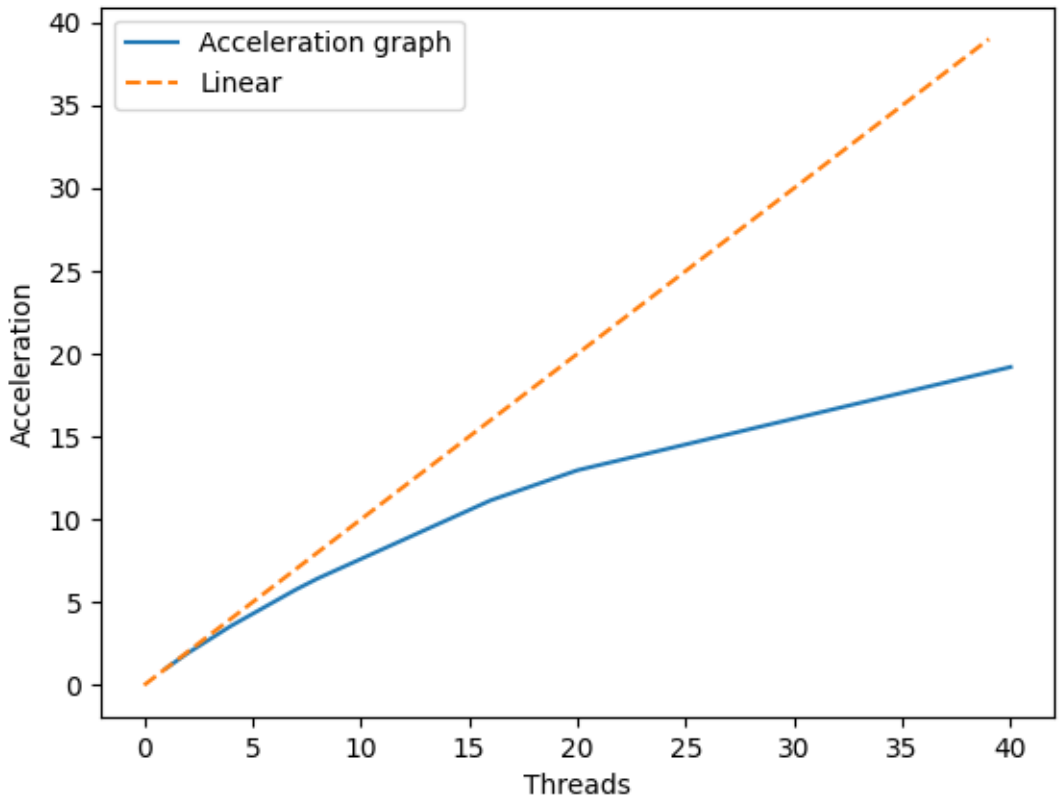
Задание 1

M = N	Количество потоков														
	2			4		7		8		16		20		40	
	T1	T2	S2	T4	S4	T7	S7	T8	S8	T16	S16	T20	S20	T40	S40
20000	1.259	0.662	1.901	0.353	3.561	0.212	5.945	0.193	6.515	0.107	11.765	0.092	13.752	0.066	18.997
40000	5.124	2.690	1.905	1.577	3.250	1.052	4.869	1.453	3.527	0.715	7.168	0.516	9.938	0.306	16.763



Вывод: по таблице и графику можно заметить, что после 16 потоков скорость выполнения программы увеличивается незначительно, а значит, смысла увеличивать количество потоков выше 40, а возможно и 20 почти нет. Также можно заметить, что программа, на которую подали больший объем данных, ускорилась меньше, чем та, на которую подали меньший. Однако, при увеличении количества потоков ситуация начала выправляться, из чего можно предположить, что при большем количестве потоков ускорение будет примерно одинаково.

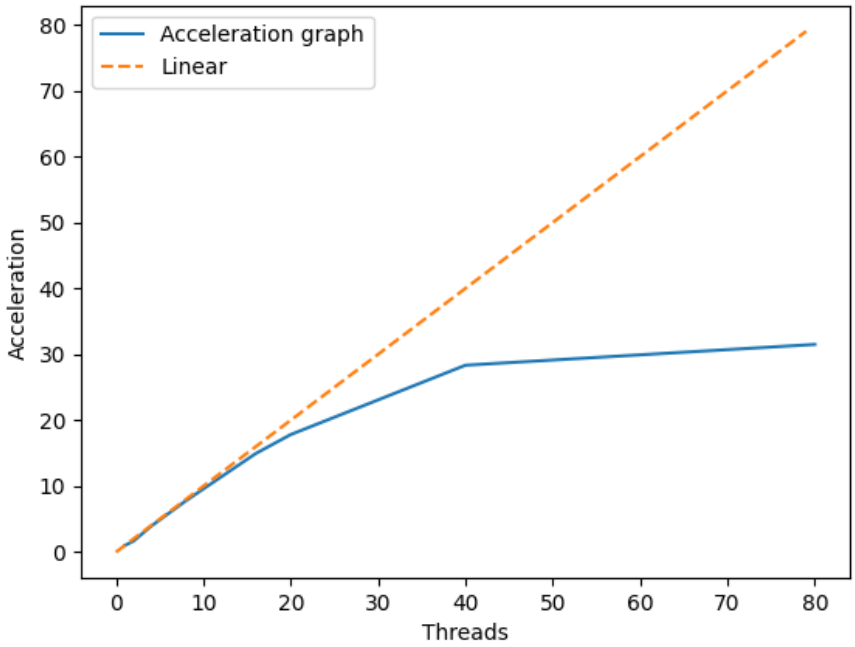
Задание 2:



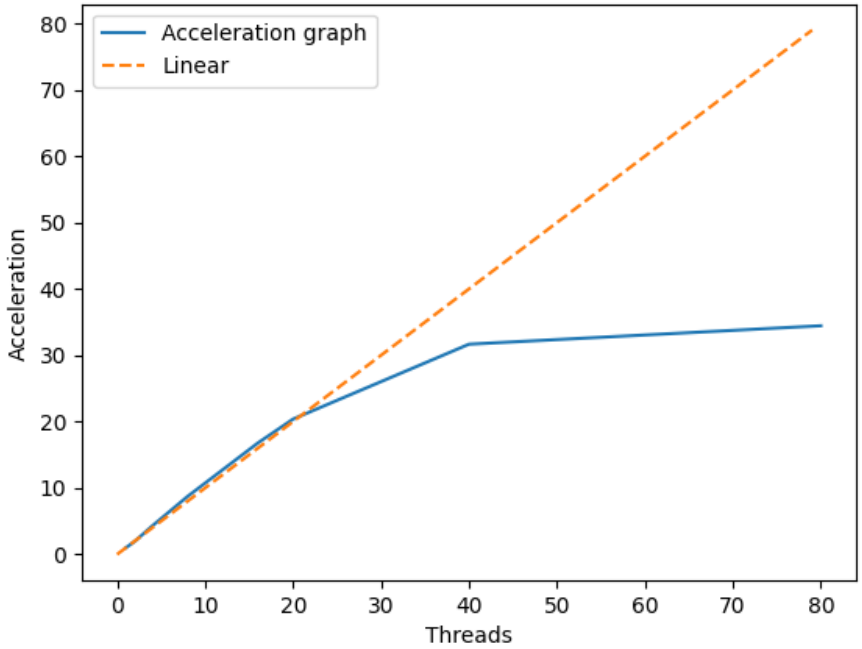
Вывод: по графику можно заметить, что ускорение работы программы при росте числа потоков стремительно падает, однако разница в ускорении между 20 и 40 потоками все еще значительна. Значит, есть смысл увеличивать количество потоков до 40, однако поднимать это значения еще выше смысла не много.

Задание 3:

Вариант 1:



Вариант 2:



Количество потоков	1	2	4	8	16	20	40	80
Вариант 1:								
Время, с	40,7	24,5	10,4	5,2	2,7	2,3	1,4	1,3
Ускорение, раз	1,0	1,7	3,9	7,8	14,9	17,8	28,3	31,5
Вариант 2:								
Время, с	38,2	19,6	8,9	4,4	2,3	1,9	1,2	1,1
Ускорение, раз	1,0	1,9	4,3	8,7	16,8	20,4	31,7	34,4

Вывод: как видно и по графикам, и по таблице. Второй вариант реализации программы (с одним omp parallel) эффективнее первого примерно на 10%.