



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ  
ТЕХНОЛОГИИ» (ИУ7)

## ДОМАШНЕЕ ЗАДАНИЕ

Дисциплина: Программирование параллельных процессов

Студент

ИУ7-31М

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Е.В. Брянская

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

А.П. Ковтушенко

(И.О. Фамилия)

Москва, 2023

## **Постановка задачи**

Разработать процедуру быстрой сортировки. Обосновать проектное решение (выбор алгоритма). Обеспечить равномерную загрузку процессоров. Результат вынести в текстовый файл построчно.

Исследовать зависимость времени счета от размерности задачи и количества процессоров.

## **Способы реализации**

Ключевые моменты алгоритма быстрой сортировки:

1. на очередном шаге выбирается опорный элемент — им может быть любой элемент массива;
2. все остальные элементы массива сравниваются с опорным и те, которые меньше него, ставятся слева от него, а которые больше или равны — справа;
3. для двух получившихся блоков массива (меньше опорного, и больше либо равны опорному) производится точно такая же операция — выделяется опорный элемент и всё идёт точно так же, пока в блоке не останется один элемент.

С привлечением MPI возможны следующие реализации данного алгоритма.

### 1 способ.

0 узел рассылает всем остальным узлам исходный массив.

Все узлы делятся на подгруппы, которые в свою очередь входят в другие подгруппы более высокого уровня, например, каждый из узлов входит в подгруппу из одного узла, которые входят в подгруппу из двух, и так далее. В каждой подгруппе выделяется лидер.

0 узел производит одну итерацию быстрой сортировки и отправляет лидеру старшей подгруппы часть массива, которая расположена правее опорной точки, после этого снова выполняет сортировку и отправку правой части массива, но уже подгруппе уровня ниже предыдущего. Так продолжается до

тех пор, пока число элементов в подгруппе не дойдёт до 1, то есть 0 узел остаётся единственным представителем, в таком случае, выполняется финальная итерация быстрой сортировки.

Аналогичные действия происходят и на других узлах, после получения своей части массива.

### 2 способ

Исходный массив разделяется на примерно равные части по числу доступных процессоров. Далее 0 узел рассылает каждому из них соответствующие им части массива.

Каждый из узлов, получив свою часть массива, производит сортировку и отправляет его 0 узлу, который сливает свой массив с тем, что получил от очередного процессора.

### 3 способ

Модификация 2 способа, отличие в том, что отсортированные части массива узлы сливают попарно, а не отправляют всё 0 узлу.

## **Результаты**

Были проведены замеры времени по следующим параметрам:

- число узлов – 1, 2, 4, 8, 10;
- размер массива – 1000, 10 000, 100 000, 1 000 000, 10 000 000, 100 000 000.

Результаты представлены в таблице ниже.

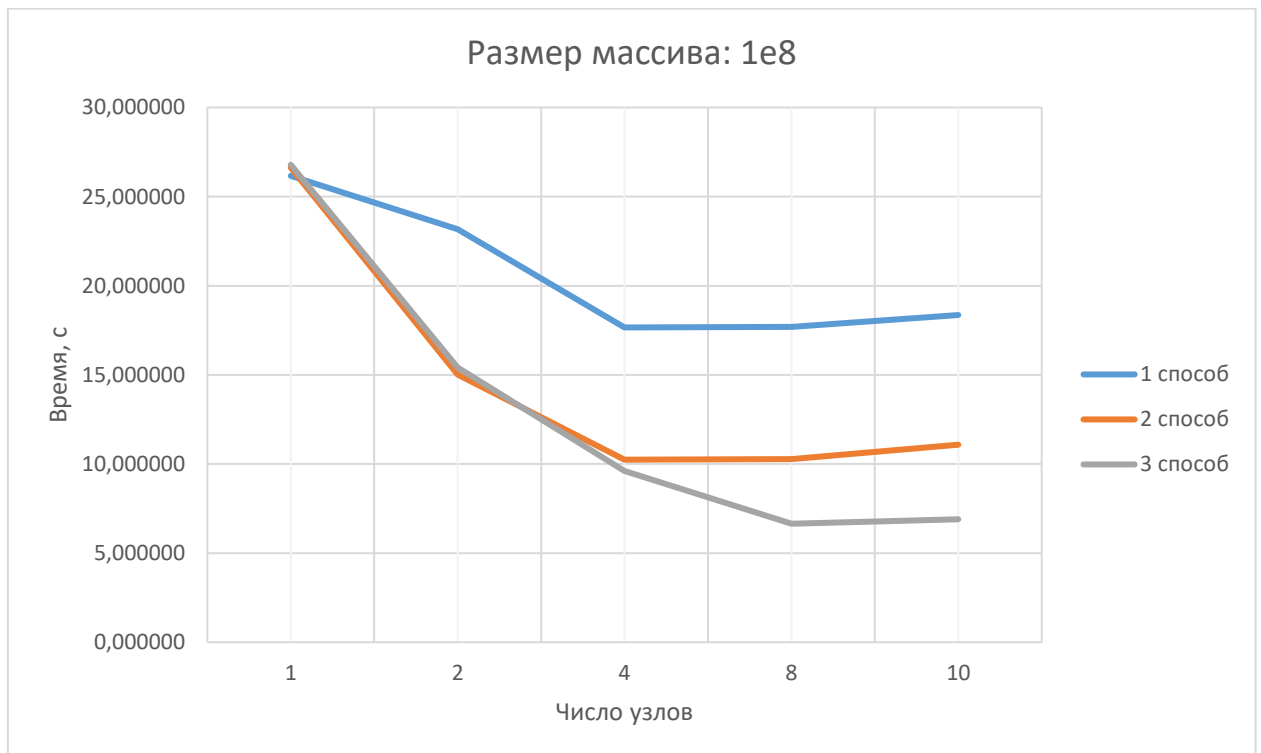
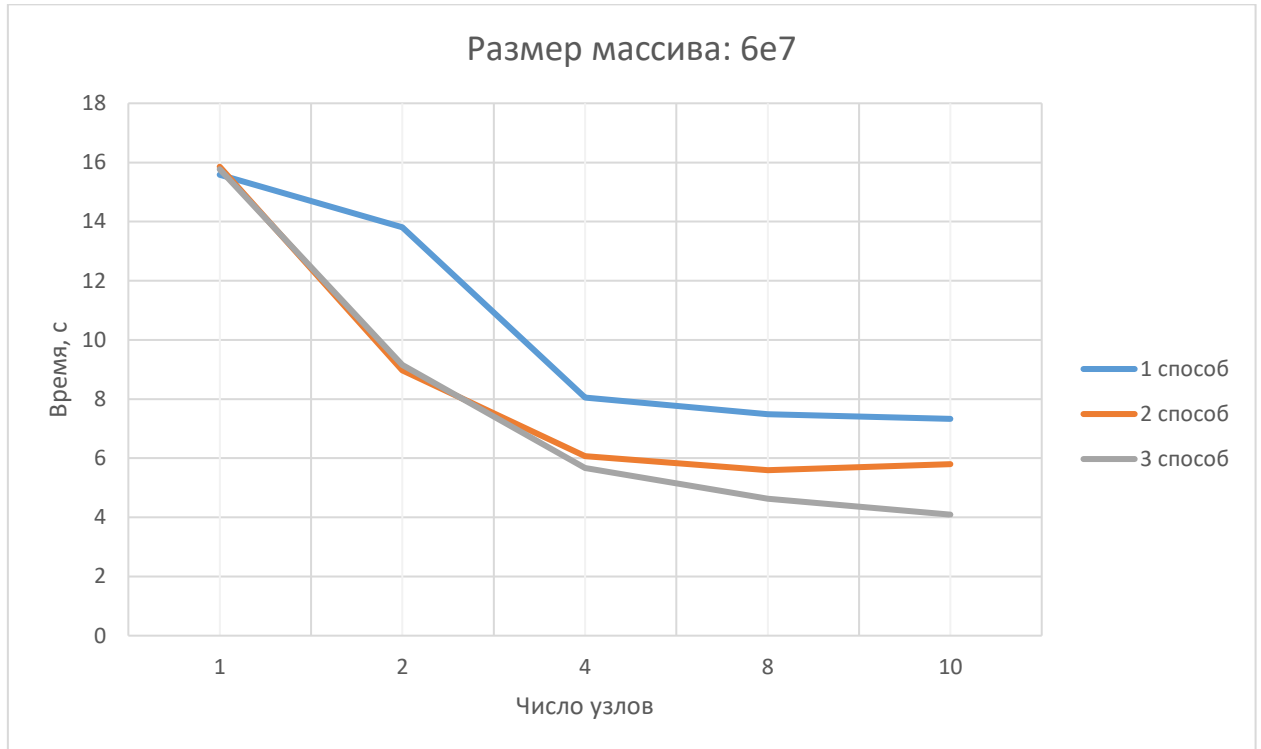
Простая реализация			1 способ Cast, итеративное разделение массива			2 способ Каждый узел по фрагменту массива, сливает 0 узел			3 способ 2 способ + попарное слияние подмассивов		
Число узлов	Размер массива	Время, с	Число узлов	Размер массива	Время, с	Число узлов	Размер массива	Время, с	Число узлов	Размер массива	Время, с
1	1,00E+03	0,000134	1	1,00E+03	0,000124	1	1,00E+03	0,000127	1	1,00E+03	0,000120
	1,00E+04	0,002508		1,00E+04	0,001556		1,00E+04	0,001544		1,00E+04	0,001499
	1,00E+05	0,018430		1,00E+05	0,018149		1,00E+05	0,018762		1,00E+05	0,018414
	1,00E+06	0,232421		1,00E+06	0,206677		1,00E+06	0,220227		1,00E+06	0,215197
	1,00E+07	2,540060		1,00E+07	2,434754		1,00E+07	2,486289		1,00E+07	2,465110
	4,00E+07	10,270959		4,00E+07	10,154075		4,00E+07	10,437589		4,00E+07	10,411373
	6,00E+07	16,228383		6,00E+07	15,584503		6,00E+07	15,855670		6,00E+07	15,770660
	1,00E+08	27,104227		1,00E+08	26,164843		1,00E+08	26,624076		1,00E+08	26,786284
			2	1,00E+03	0,011356	2	1,00E+03	0,011378	2	1,00E+03	0,011318
				1,00E+04	0,012510		1,00E+04	0,012133		1,00E+04	0,012866
				1,00E+05	0,024822		1,00E+05	0,020749		1,00E+05	0,021212
				1,00E+06	0,195171		1,00E+06	0,134922		1,00E+06	0,138236
				1,00E+07	2,191241		1,00E+07	1,433813		1,00E+07	1,467939
				4,00E+07	8,935597		4,00E+07	5,941506		4,00E+07	6,114605
				6,00E+07	13,811556		6,00E+07	8,974751		6,00E+07	9,157403
				1,00E+08	23,165154		1,00E+08	15,014164		1,00E+08	15,420049
			4	1,00E+03	0,021881	4	1,00E+03	0,031686	4	1,00E+03	0,032779
				1,00E+04	0,021862		1,00E+04	0,032034		1,00E+04	0,033631
				1,00E+05	0,030377		1,00E+05	0,038440		1,00E+05	0,037782
				1,00E+06	0,165307		1,00E+06	0,114217		1,00E+06	0,109340
				1,00E+07	2,069434		1,00E+07	1,017964		1,00E+07	0,958958
				4,00E+07	9,227727		4,00E+07	4,159074		4,00E+07	3,787686
				6,00E+07	8,046086		6,00E+07	6,069553		6,00E+07	5,663826
				1,00E+08	17,661725		1,00E+08	10,242729		1,00E+08	9,610553
			8	1,00E+03	0,032696	8	1,00E+03	0,072052	8	1,00E+03	0,073569
				1,00E+04	0,034939		1,00E+04	0,071992		1,00E+04	0,074436
				1,00E+05	0,032759		1,00E+05	0,076626		1,00E+05	0,078705
				1,00E+06	0,106812		1,00E+06	0,152238		1,00E+06	0,139428
				1,00E+07	1,969279		1,00E+07	0,949020		1,00E+07	0,712671
				4,00E+07	5,612908		4,00E+07	3,711682		4,00E+07	3,518538

			6,00E+07	7,487763			6,00E+07	5,594829			6,00E+07	4,626111
			1,00E+08	17,696990			1,00E+08	10,281099			1,00E+08	6,648949
		10	1,00E+03	0,033358		10	1,00E+03	0,091995		10	1,00E+03	0,092632
			1,00E+04	0,033279			1,00E+04	0,091553			1,00E+04	0,092625
			1,00E+05	0,040896			1,00E+05	0,096811			1,00E+05	0,096196
			1,00E+06	0,112668			1,00E+06	0,180145			1,00E+06	0,145656
			1,00E+07	1,489346			1,00E+07	0,981820			1,00E+07	0,726073
			4,00E+07	5,427860			4,00E+07	3,862866			4,00E+07	2,855161
			6,00E+07	7,329851			6,00E+07	5,800010			6,00E+07	4,091891
			1,00E+08	18,356675			1,00E+08	11,087198			1,00E+08	6,893851

## Графики зависимостей

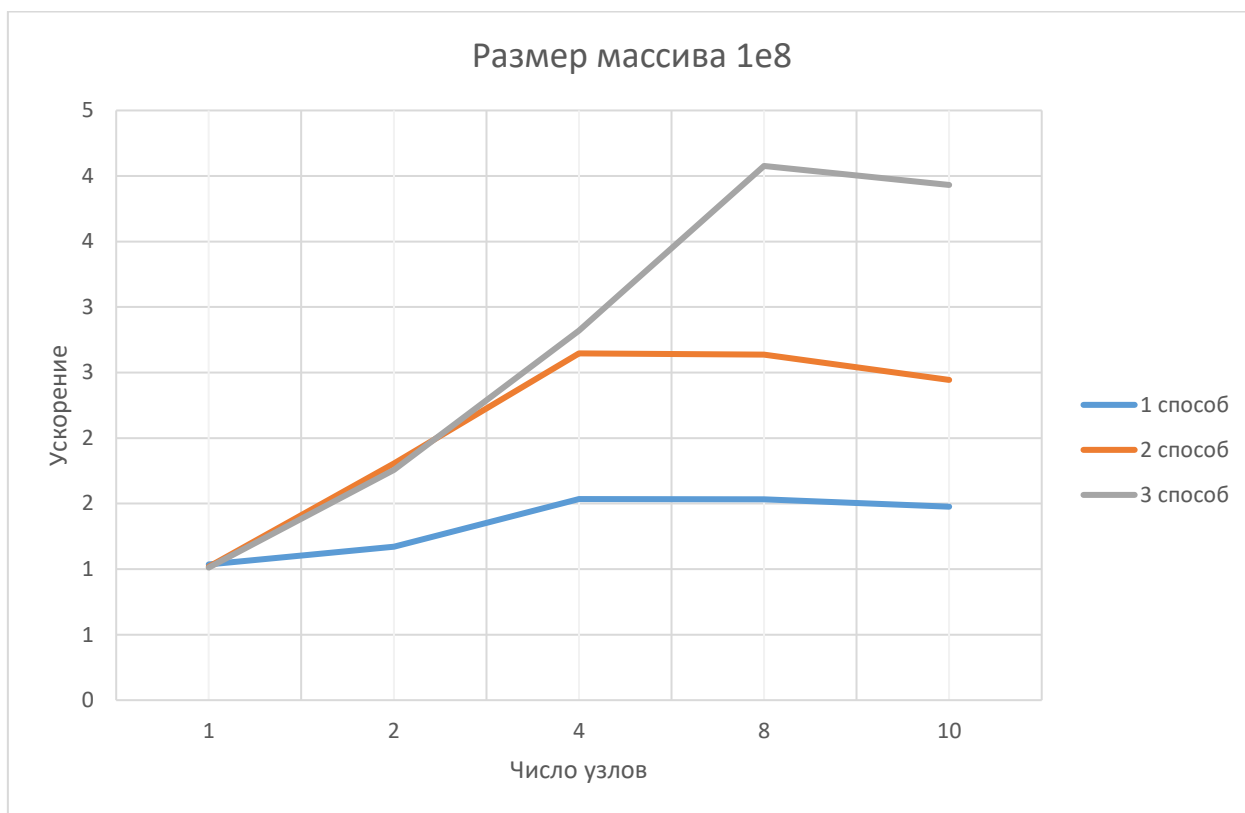
### Зависимость времени от количества узлов

Исходя из замеров, можно сделать вывод о том, что целесообразно исследовать показатели только на больших размерах массива. Ниже приведено два графика для  $6e7$  и  $1e8$  элементов.

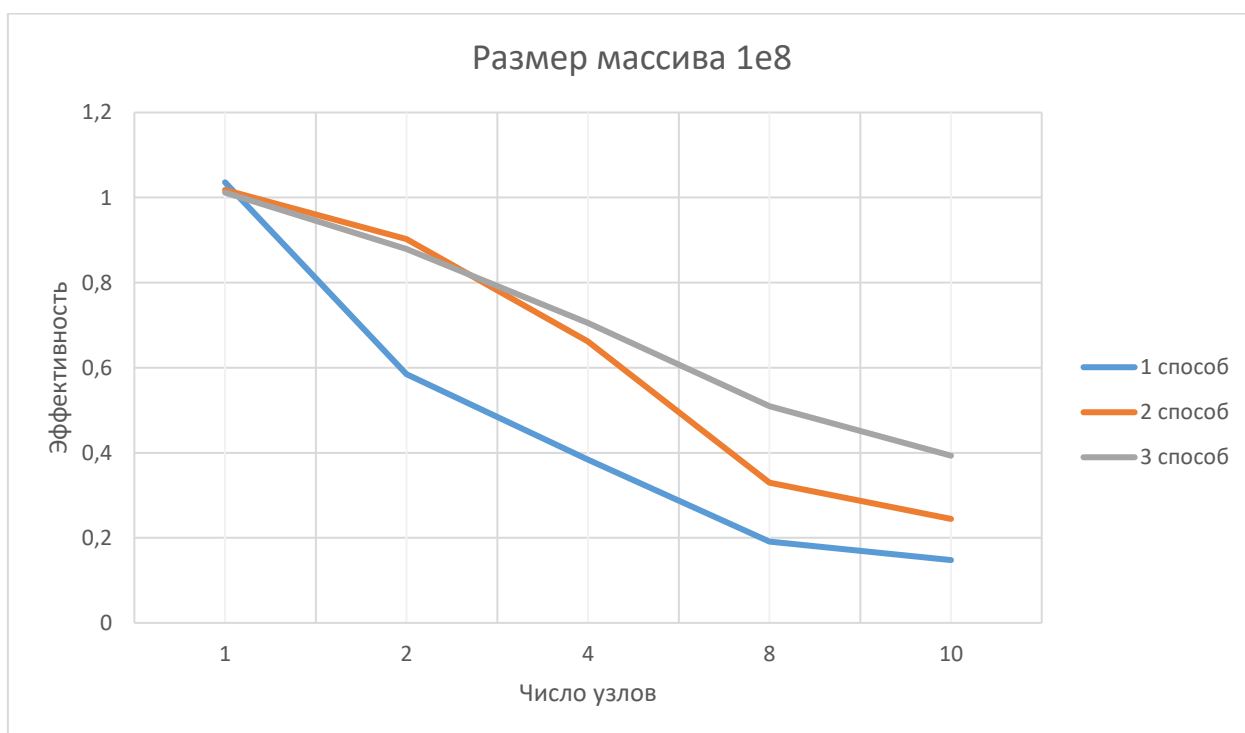


Далее будут приведены расчёты только для размера  $1e8$ .

### Ускорение от числа узлов



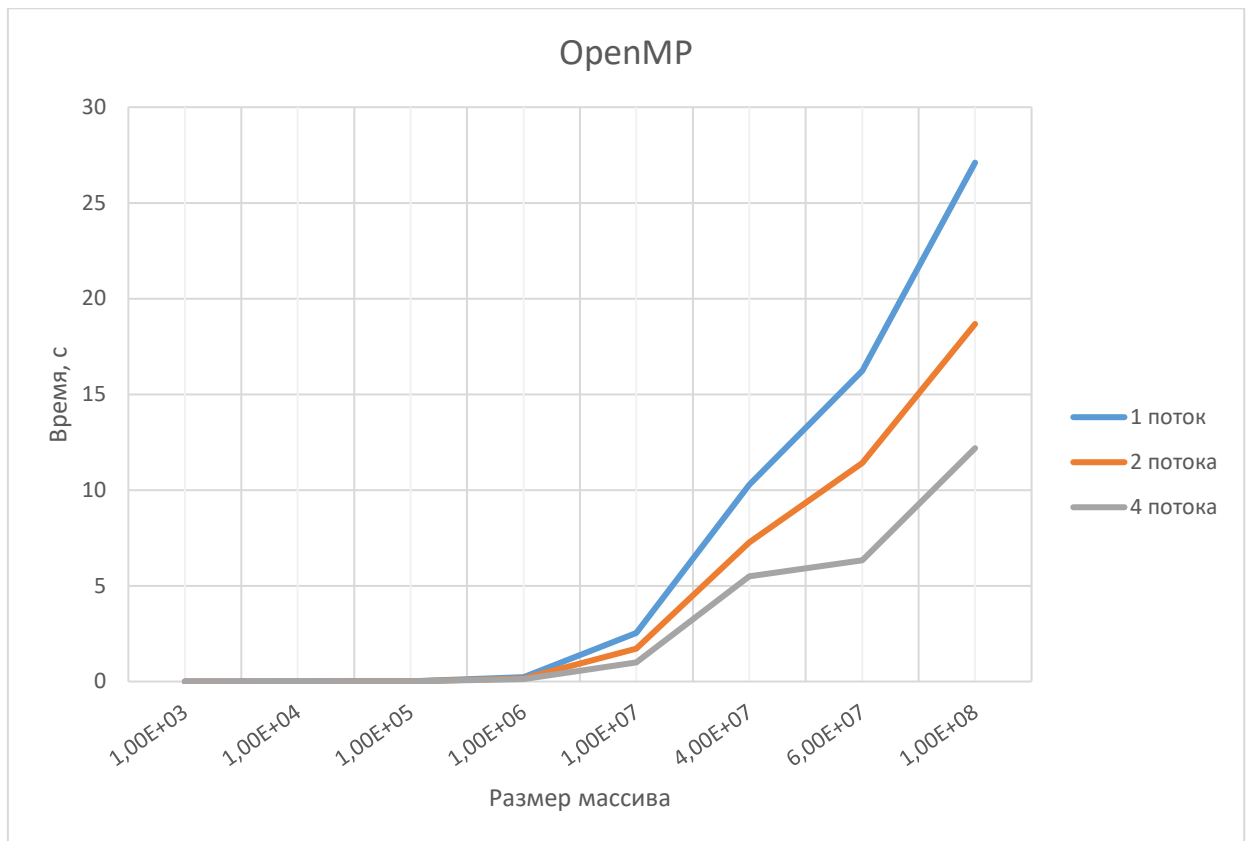
### Эффективность



## OpenMP

Число потоков	Размер массива	Время, с
1	1,00E+03	0,000134
	1,00E+04	0,002508
	1,00E+05	0,018430
	1,00E+06	0,232421
	1,00E+07	2,540060
	4,00E+07	10,270959
	6,00E+07	16,228383
	1,00E+08	27,104227
2	1,00E+03	0,000271
	1,00E+04	0,001240
	1,00E+05	0,013231
	1,00E+06	0,163178
	1,00E+07	1,711063
	4,00E+07	7,268493
	6,00E+07	11,413092
	1,00E+08	18,678325
4	1,00E+03	0,000548
	1,00E+04	0,001280
	1,00E+05	0,009520
	1,00E+06	0,120951
	1,00E+07	1,005451
	4,00E+07	5,502056
	6,00E+07	6,338169
	1,00E+08	12,188033





Ускорение и эффективность рассчитываются для массива в  $1e8$  элементов.

Число потоков	Ускорение
1	1
2	1,45110587
4	2,22383932

Число потоков	Эффективность
1	1
2	0,72555293
4	0,55595983

Далее все замеры производятся с привлечением 4 потоков и только числе узлов равно 1, 2, 4, т.к. на момент проведения замеров 3 из 10 узлов были не доступны.

Число узлов	Размер массива	Время, с
1	1,00E+03	0,000343
	1,00E+04	0,001049
	1,00E+05	0,010519
	1,00E+06	0,105294
	1,00E+07	1,349855

		4,00E+07	4,689517
		6,00E+07	7,073439
		1,00E+08	11,042553
2		1,00E+03	0,012162
		1,00E+04	0,012659
		1,00E+05	0,016407
		1,00E+06	0,076262
		1,00E+07	1,060716
		4,00E+07	3,573357
		6,00E+07	7,454172
		1,00E+08	9,235273
4		1,00E+03	0,032819
		1,00E+04	0,033342
		1,00E+05	0,037332
		1,00E+06	0,091422
		1,00E+07	0,765110
		4,00E+07	3,215622
		6,00E+07	5,132323
		1,00E+08	7,937487