Государственное образовательное учреждение высшего профессионального образования
«Московский государственный технический университет
имени Н. Э. Баумана»
ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ
По лабораторной работе №3
По курсу "Анализ алгоритмов"
Тема: "Исследование сложности алгоритмов сортировки"

Студент: Бадалян Д.А.
Группа: ИУ7-51

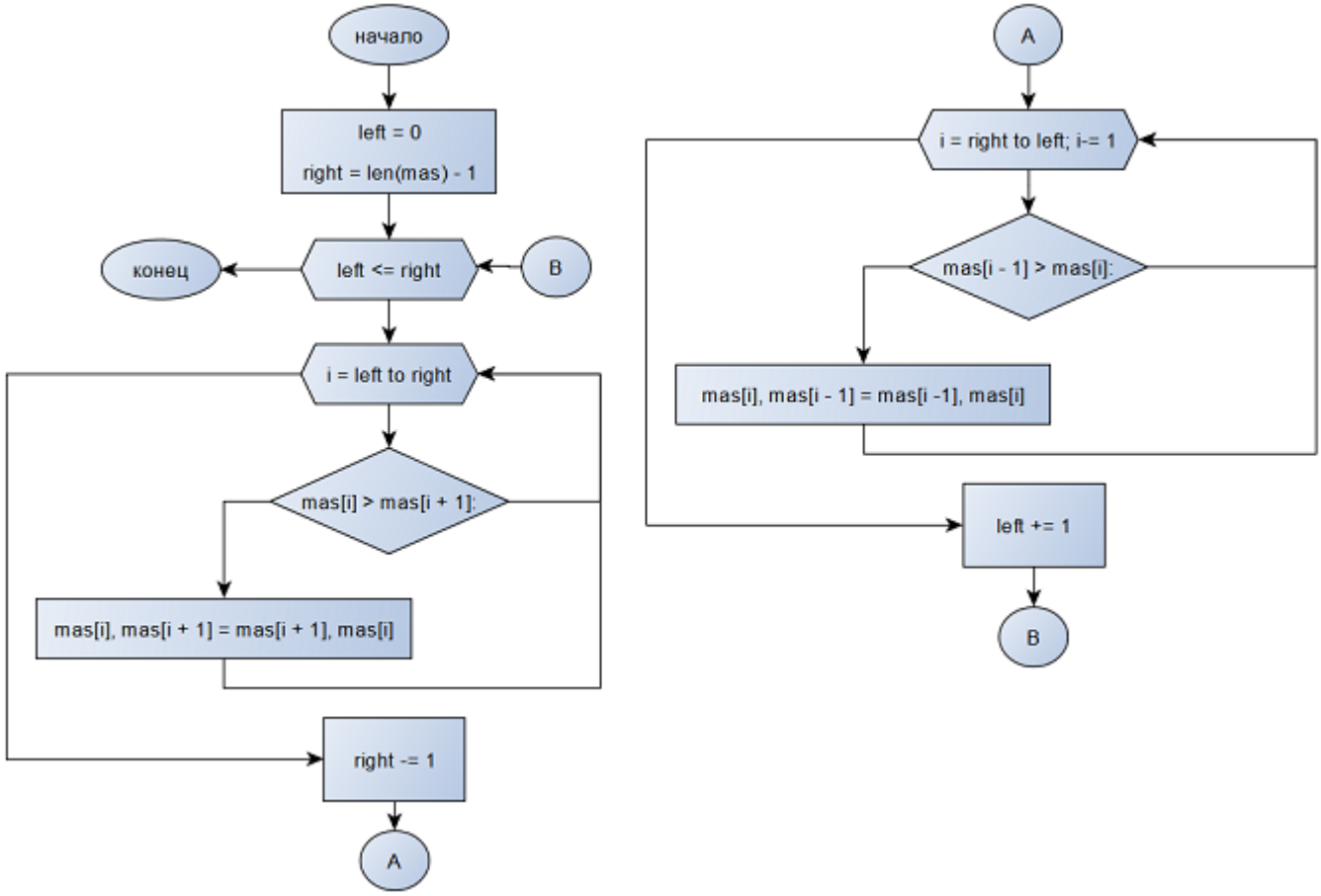
Москва, 2017

Постановка задачи

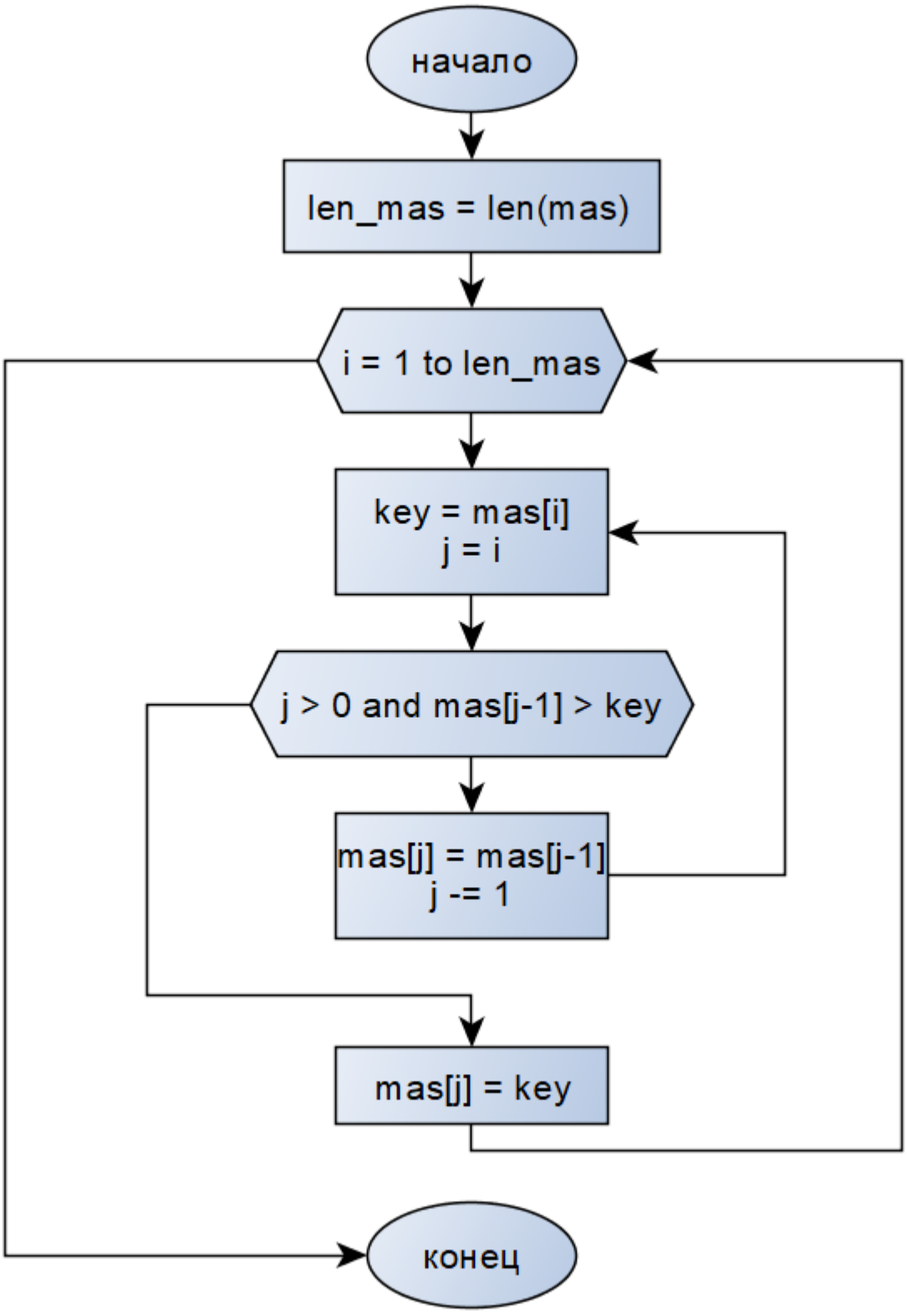
Реализовать алгоритм сортировки:
1. Шейкер сортировка;
2. Сортировка вставками;
3. Сортировка выбором;
Рассчитать трудоёмкость алгоритмов и провести временные замеры.

Алгоритмы

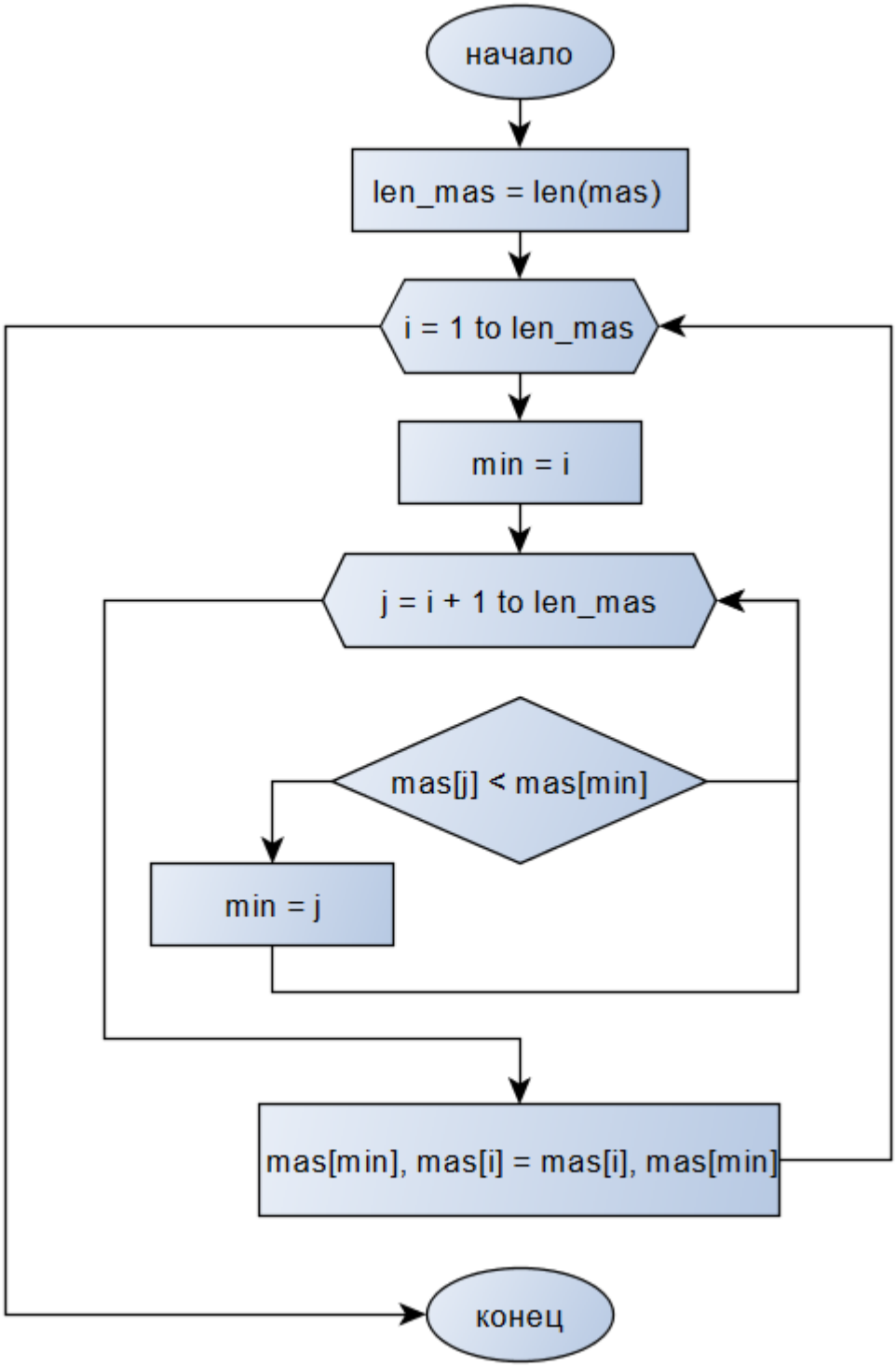
Шейкер сортировка:



Сортировка вставками:



Сортировка выбором:



Трудоёмкость

Модель вычисления:
1. вызов метода объекта класса имеет трудоёмкость 1;
2. объявление переменной/массива/структуры без определения имеет трудоёмкость 0;
3. оператор $+, -, *, /, =, ++, -$ имеют трудоёмкость 1;
4. условный оператор (без условий внутри) имеет трудоёмкость 0;
5. логические операции имеют трудоёмкость 1;
6. трудоёмкость цикла for $2 + n * (2 + T)$, где n - это число повторений цикла, T - трудоёмкость цикла;
7. трудоёмкость цикла while $1 + n * (1 + U + T)$, где n - это число повторений цикла, T - трудоёмкость цикла, U - условия, задаваемые в цикле while;

Расчет трудоёмкости:
Обозначения: n - длина массива

Шейкер сортировка:
$F1best = 12n^2 + 7n + 5$
$F1worst = 28n^2 + 7n + 5$

Сортировка вставками:
$F2avg = N^2/4$
$F2best = 10n - 8$
$F2worst = N^2/2$

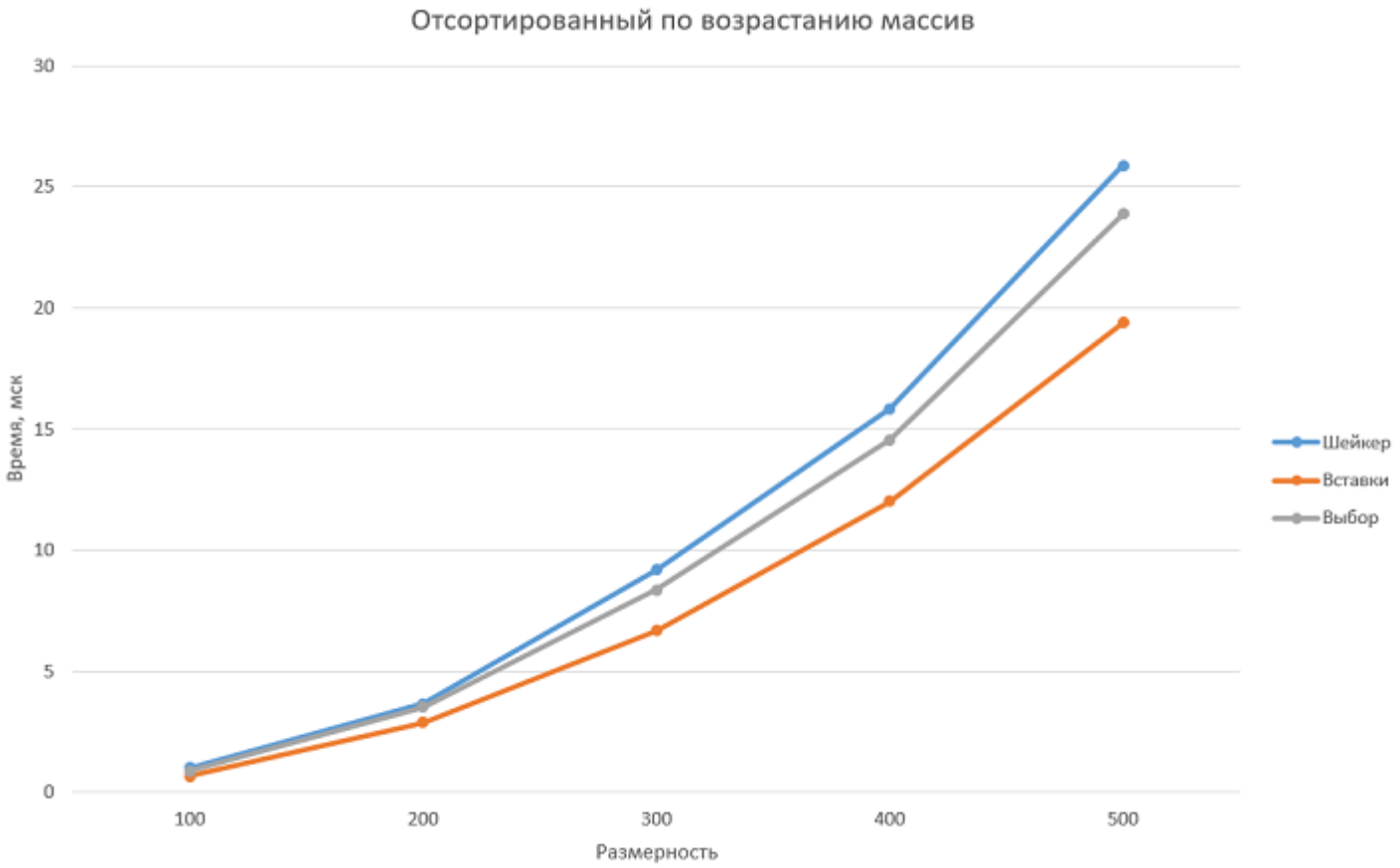
Сортировка выбором:
$F3best = 5n^2 + 11n + 3$
$F3worst = 6n^2 + 11n + 3$

Время

Замеры времени проводились на языке Python с помощью функции clock() из библиотеки time. Время указано в милисекундах.
Для замеров времени сформировано 4 массива:
1. отсортированный по возрастанию;
2. отсортированный по убыванию;
3. заполненный случайными числами от -1000 до 1000;
4. заполненный совпадающими числами (1...1);

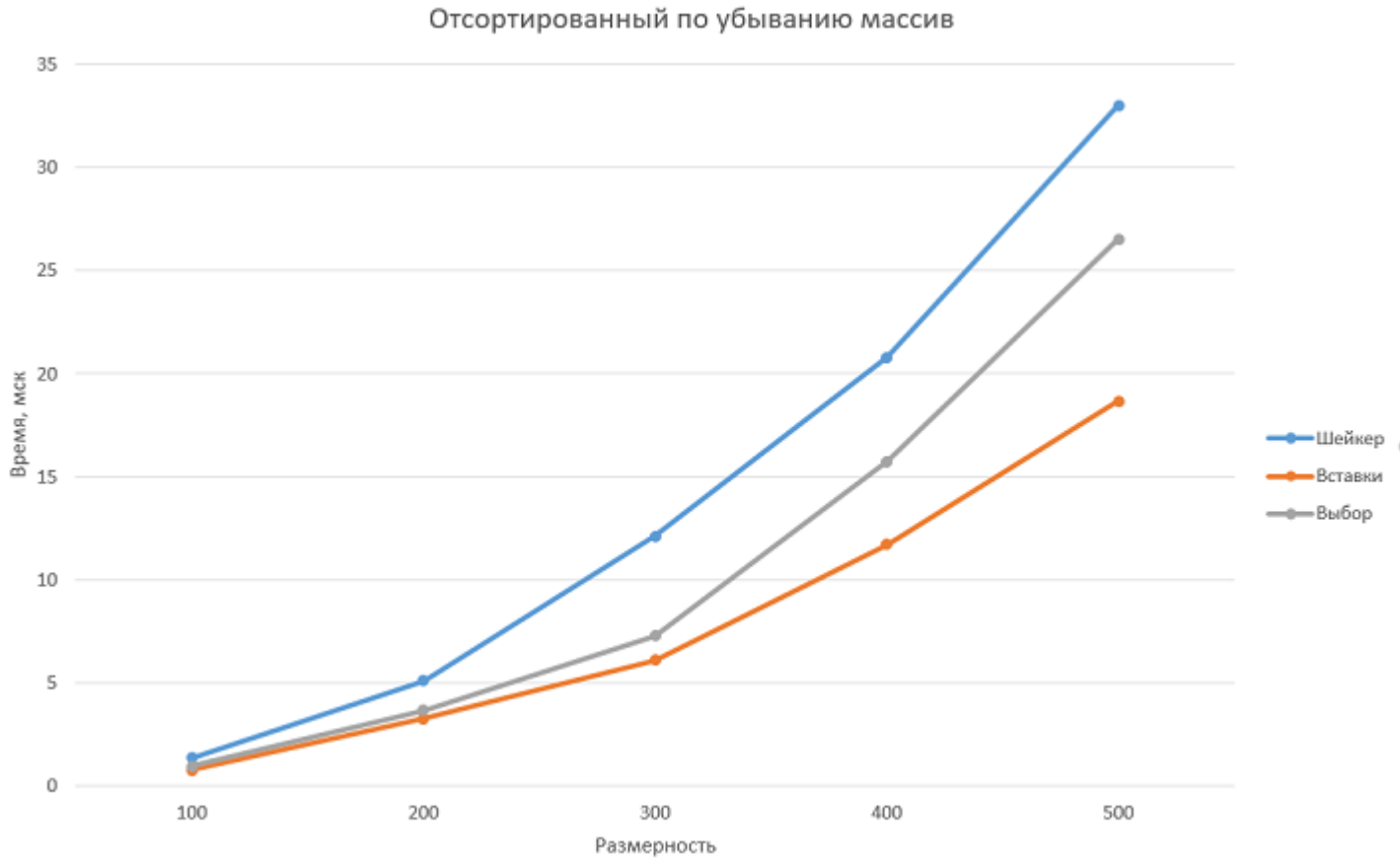
Временные замеры для массива 1:

Размерность	Шейкер	Вставки	Выбор
100	1	0.65	0,87
200	4.67	2,894	3,534
300	9,175	6,678	8,353
400	15,843	12,012	14,566
500	25,89	19,405	23,873



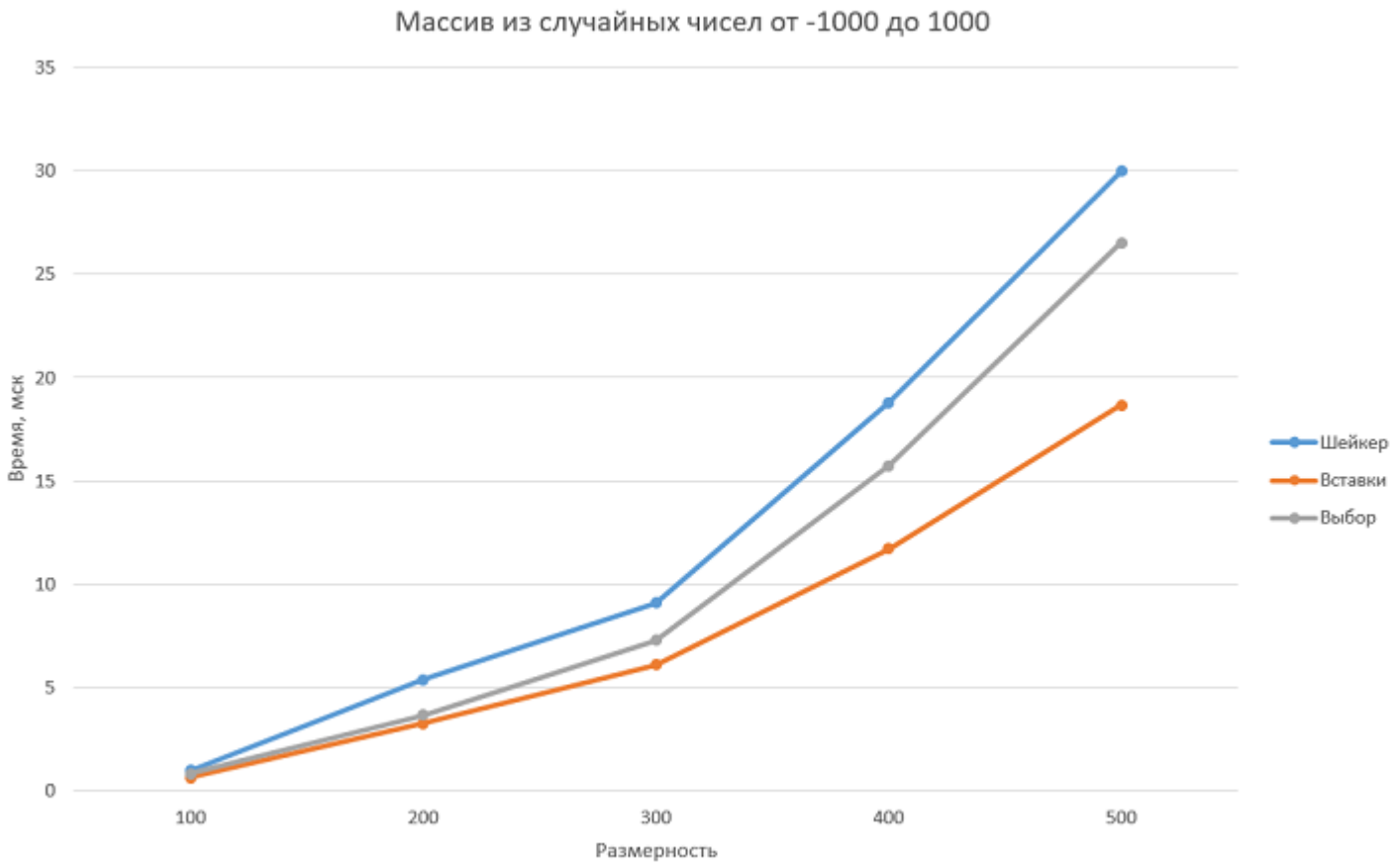
Временные замеры для массива 2:

Размерность	Шейкер	Вставки	Выбор
100	1,3804	0.86	0,96
200	5,27	3,27	3,68
300	12,108	6,103	7,282
400	20,787	11,715	15,726
500	32,997	18,661	22,529



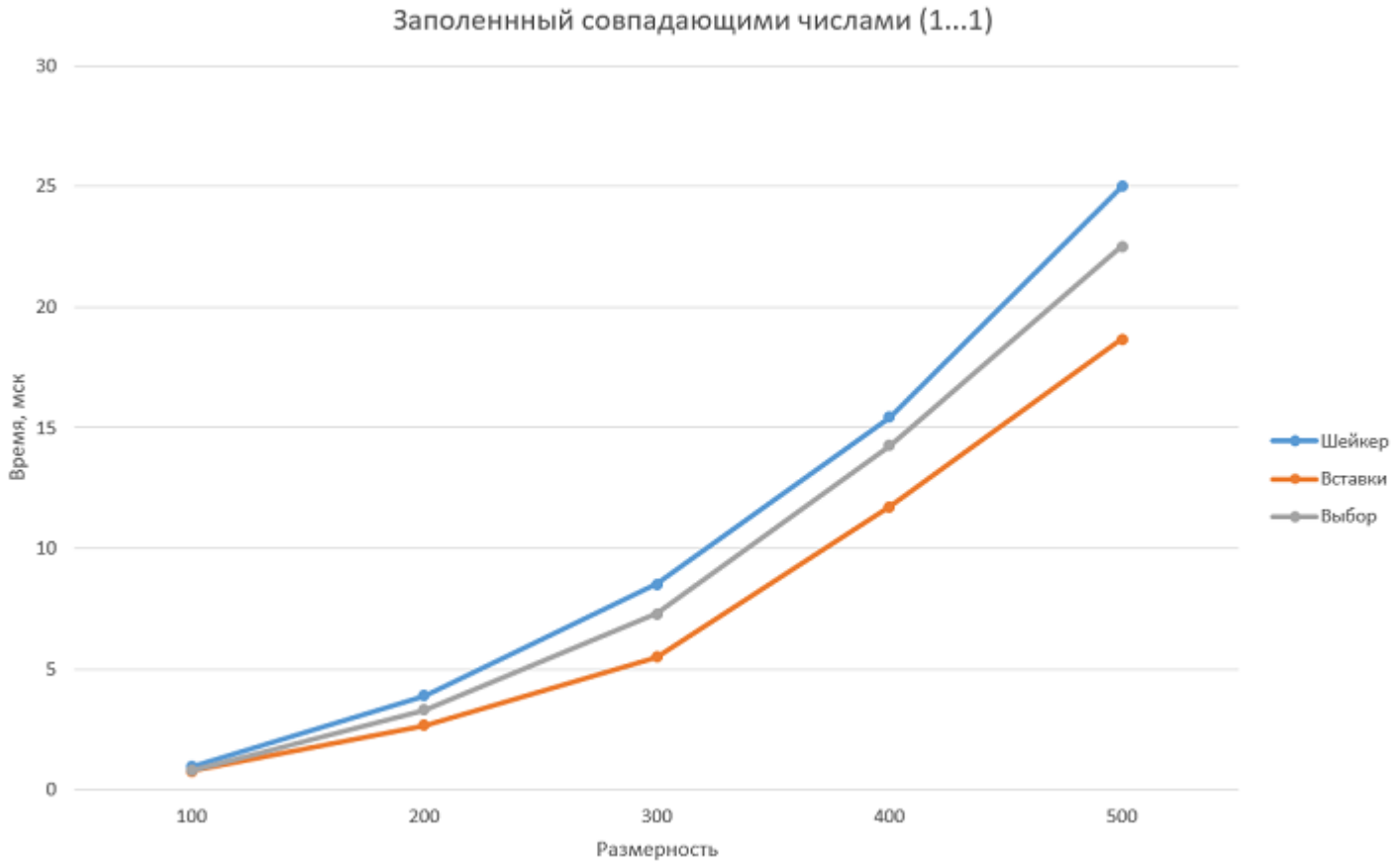
Временные замеры для массива 3:

Размерность	Шейкер	Вставки	Выбор
100	0,9802	0.76	0,83
200	4.34	3,27	3,68
300	9,108	6,103	7,282
400	18,787	11,715	15,726
500	29,997	18,661	26,529



Временные замеры для массива 4:

Размерность	Шейкер	Вставки	Выбор
100	0,964	0.76	0,8
200	3,67	2,67	3,32
300	8,53	5,5	7,282
400	15,45	11,715	14,726
500	24,997	18,661	22,529



Вывод

Шейкер сортировка, являющаяся модификацией сортировки пузырьком, показала себя хуже всего. В этом можно убедиться и при анализе рассчитанных трудоёмкостей алгоритмов. Сортировка вставками оказалась самой быстрой из рассмотренных. Так же стоит отметить, что в лучшем случае её трудоёмкость сводиться к O(n)(лучшем случаем является упорядоченный массив). Сортировка выбором, в отличие от двух других, имеет самый маленький разброс между худшим и лучшим случаями.