Сортировки. Отчёт.

1. Цель работы:

Экспериментальным путём выяснить время работы различных сортировочных алгоритмов на различных данных.

1. Параметры вычислительного узла:
   1. Процессор: Intel Core i5-3470 3.20 GHz
   2. Память: 8GB
   3. ОС: Windows 7
2. Тестируемые алгоритмы:
   1. Сортировка пузырьком

Алгоритм состоит в повторяющихся проходах по массиву, во время каждого из которых самый лёгкий (==максимальный) элемент всплывает наверх. За каждый проход элементы попарно сравниваются, и, если стоят на неверных местах, меняются местами.

* 1. Сортировка слиянием

Сортируемый массив разбивается на две части (обычно пополам), каждая из частей сортируется отдельно: разбивается на две части, и так далее. Массив длины 1 считается упорядоченным. Затем происходит merge: на каждом шаге мы берём меньший из двух первых элементов отсортированных подмассивов и записываем его в результирующий массив. Когда один из подмассивов закончился, мы добавляем все оставшиеся элементы второго подмассива в результирующий массив.

* 1. Сортировка Шелла

Действует практически как сортировка пузырьком, только, в то время как во второй сравниваются соседние элементы, в этой сравниваются элементы, находящиеся на определённом расстоянии друг от друга.

* 1. Сортировка Хоара (быстрая сортировка)

Выбирается произвольный опорный элемент, затем, все элементы равные или большие его перемещаются направо, а меньшие – налево. Затем те же самые действия применяются к получившимся двум подмассивам.

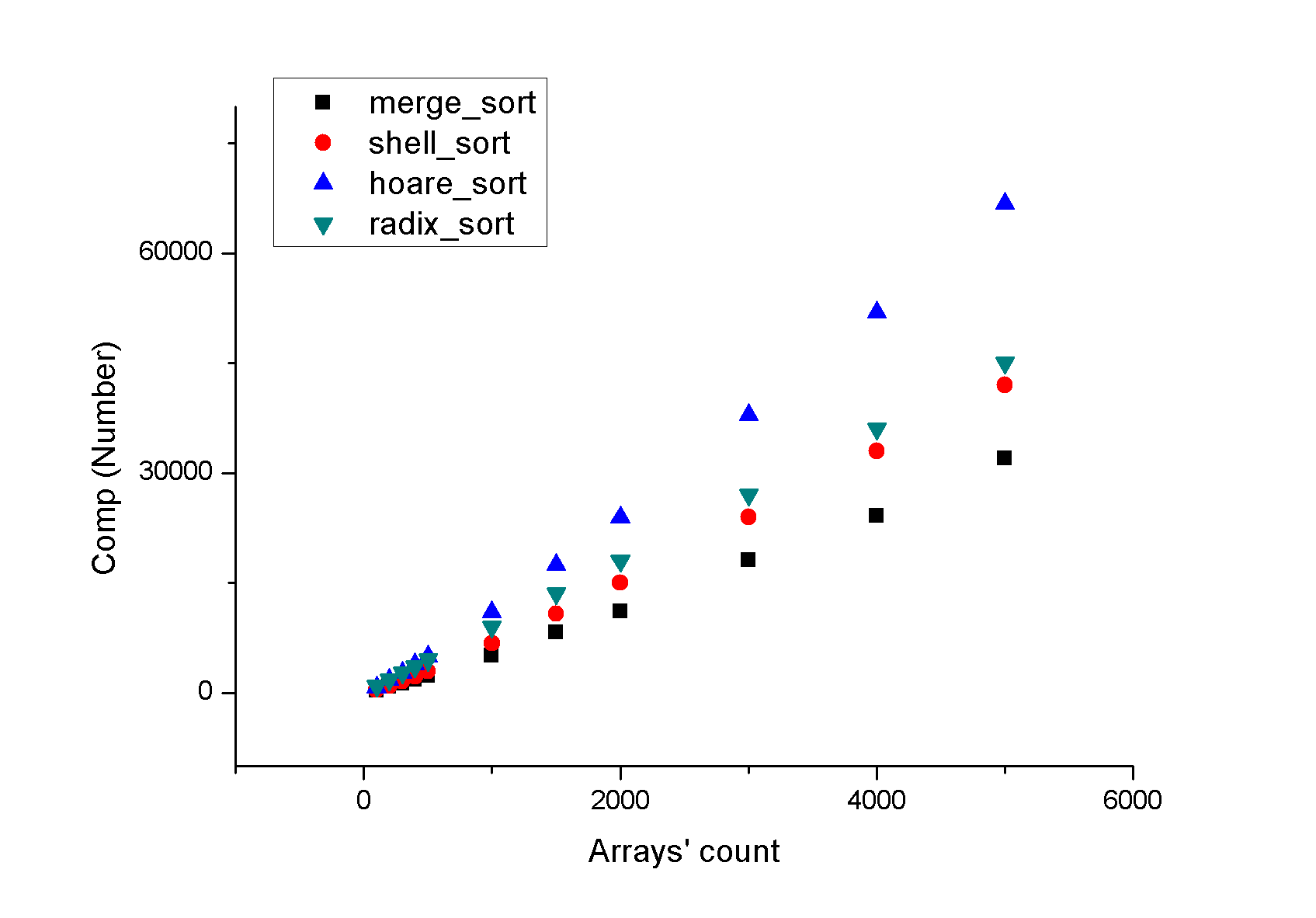
* 1. Поразрядная сортировка (least significant digit (LSD))

Алгоритм создан для сортировки целых чисел, путём разделения их на группы с одинаковыми с цифрами в каждом из разрядов.

1. Результаты измерений:

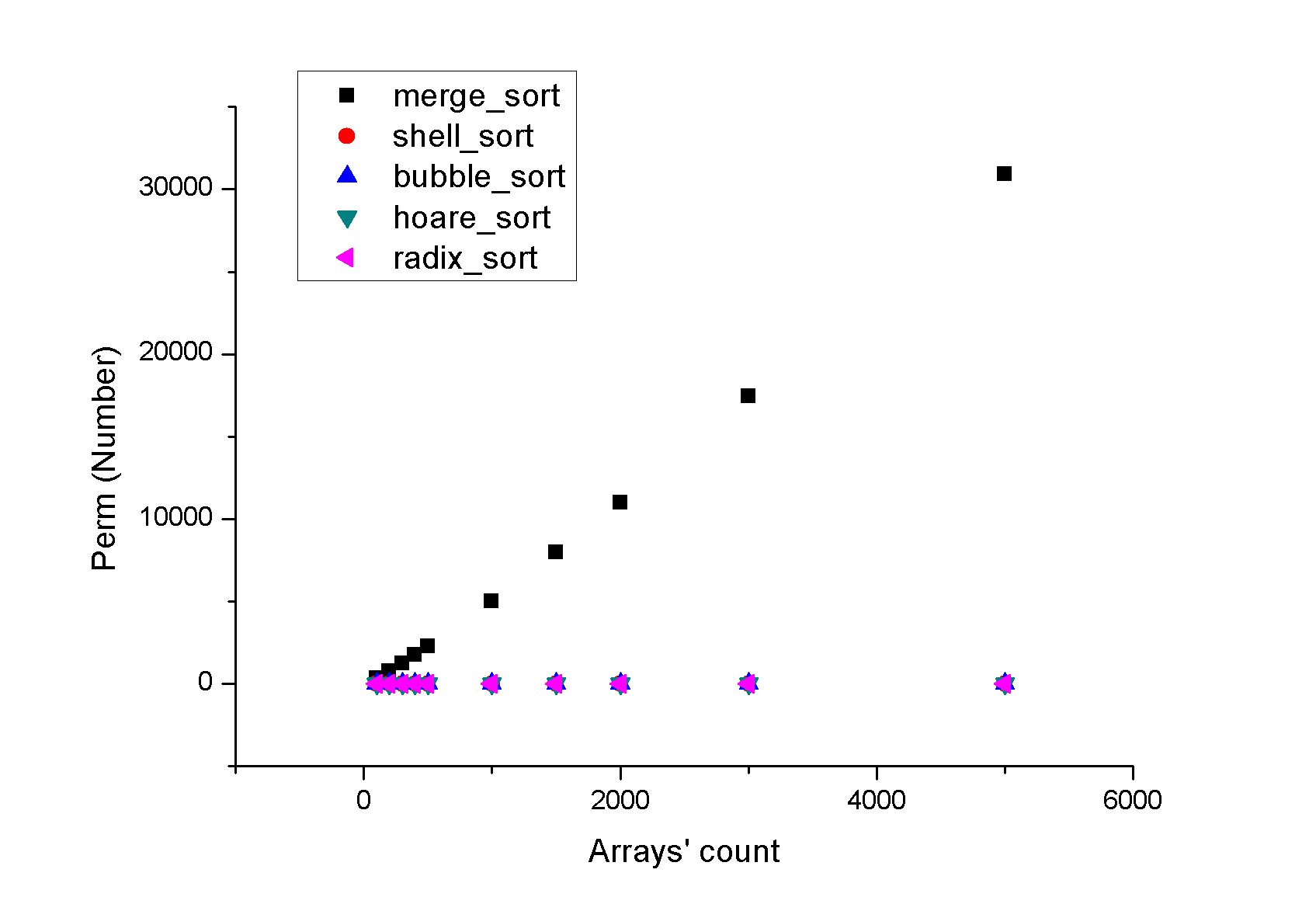
*Большинство графиков – без пузырьковой сортировки, чтобы лучше был виден масштаб между остальными алгоритмами. Графики с пузырьковой сортировкой – в папке Отчёт->Графики*

* 1. **Лучший случай:**
     1. Сравнения:

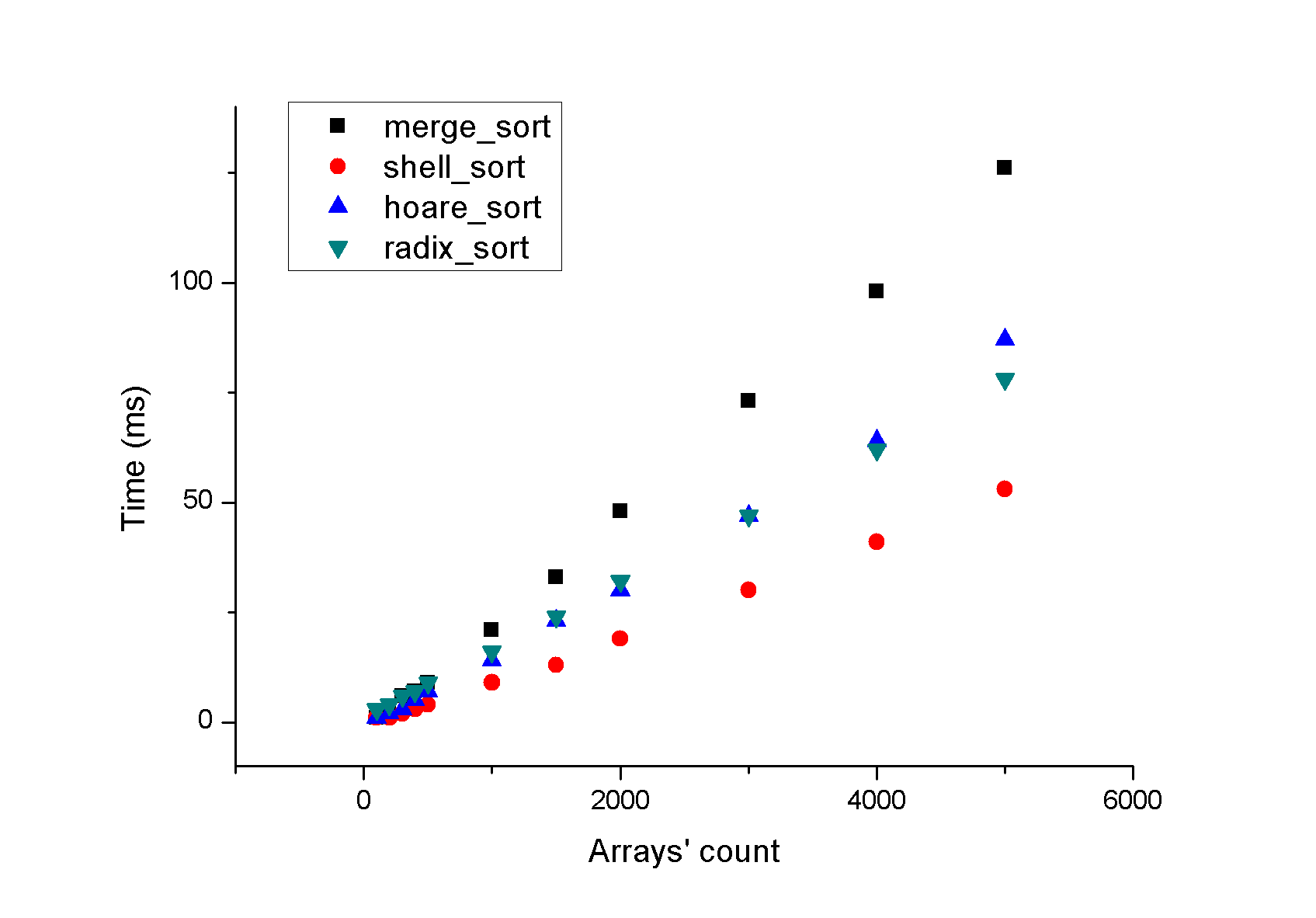


* + 1. Перестановки

*Для поразрядной сортировки невозможно посчитать в связи с реализацией.*



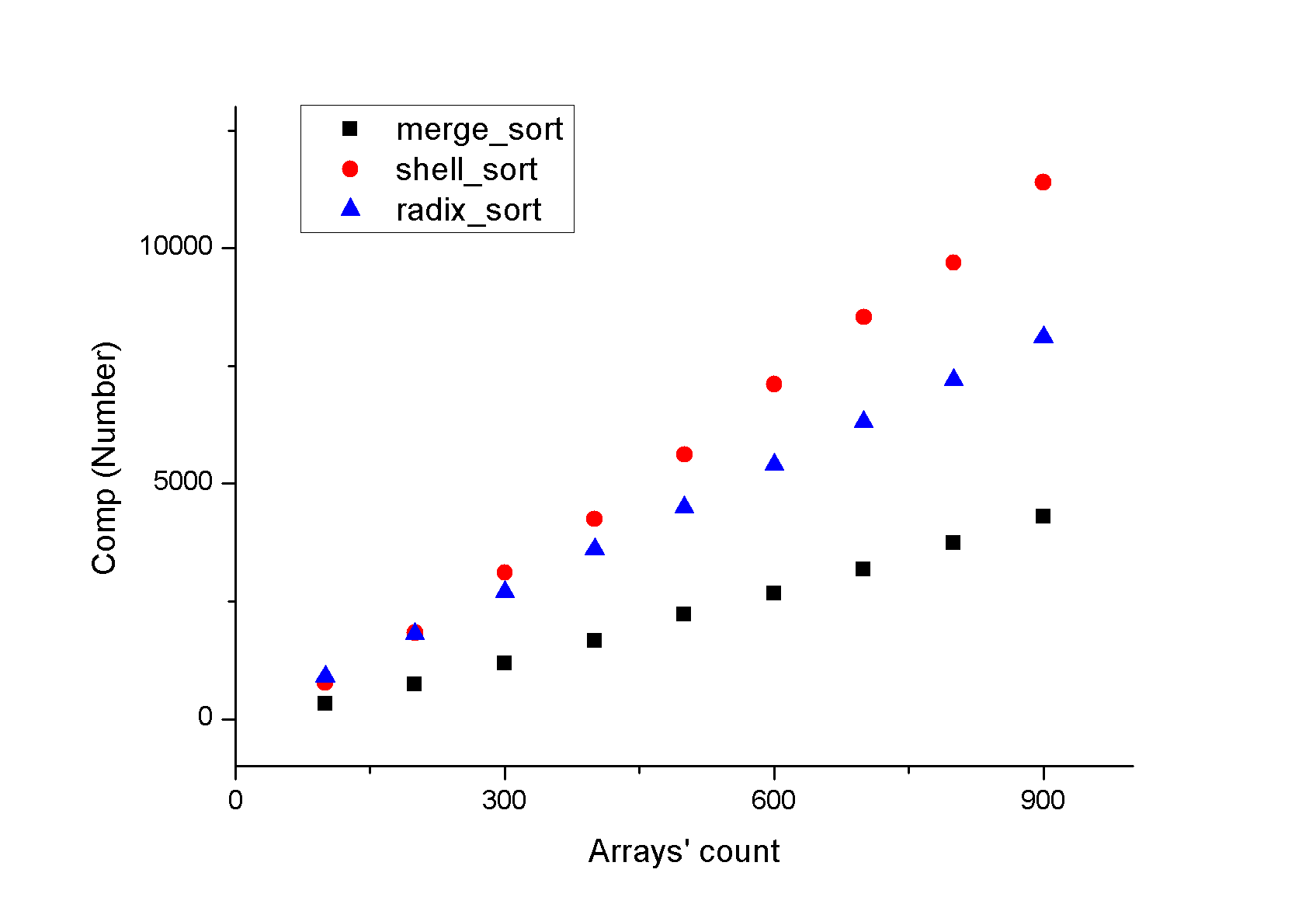
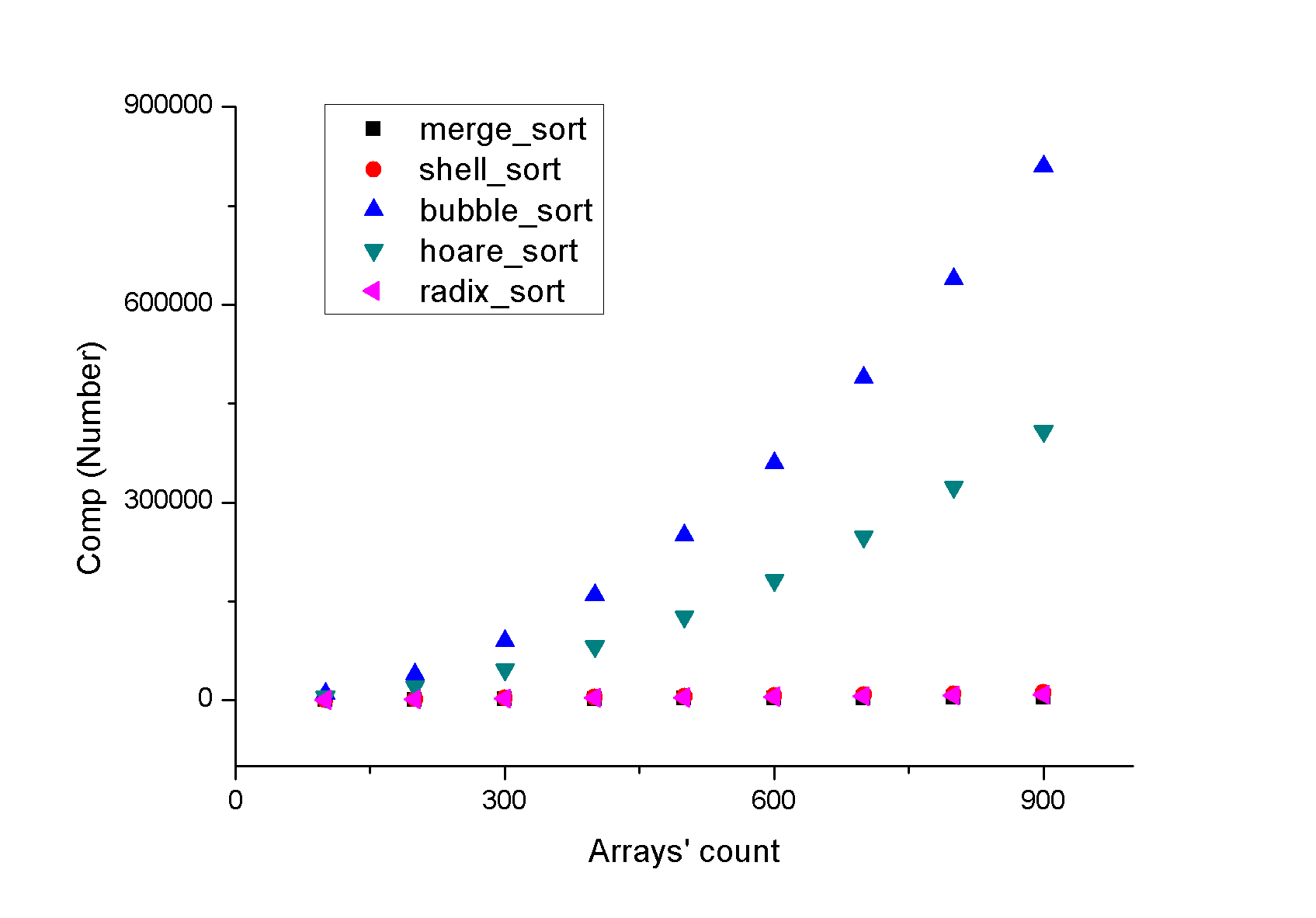
* + 1. Время



* 1. **Худший случай:**

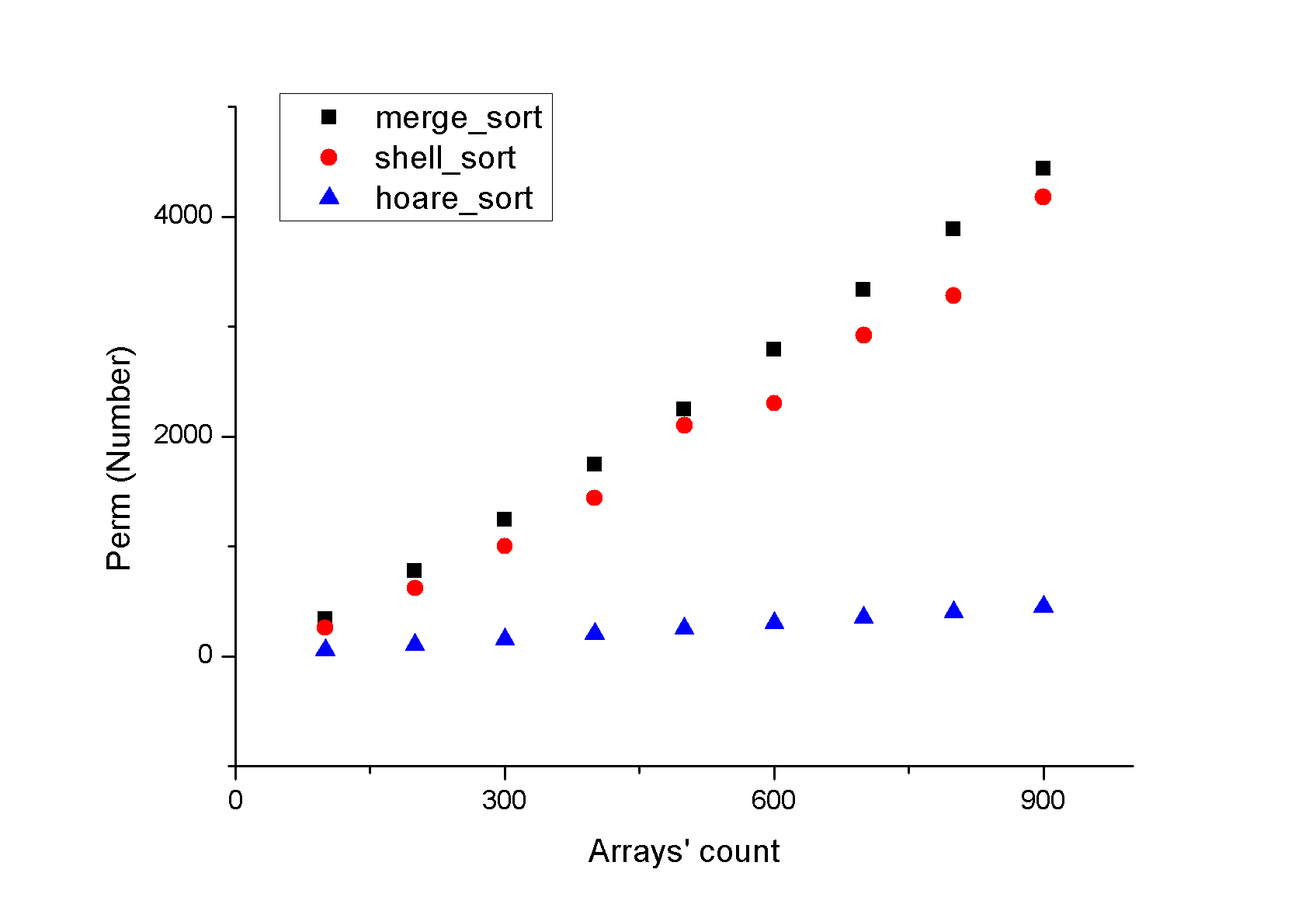
*При n=1000 сортировка Хоара достигает максимальной глубины рекурсии и падает.*

* + 1. Сравнения:

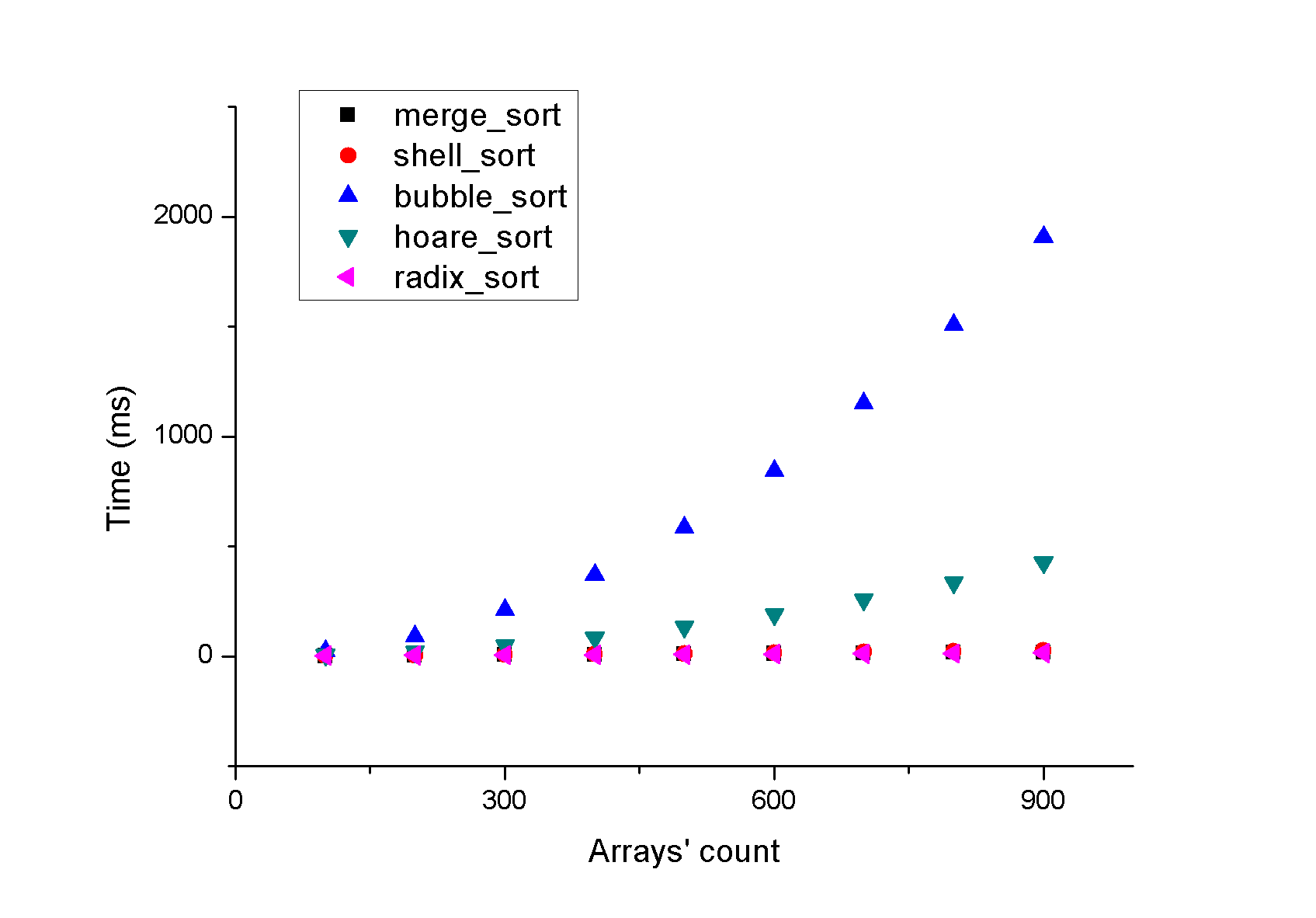


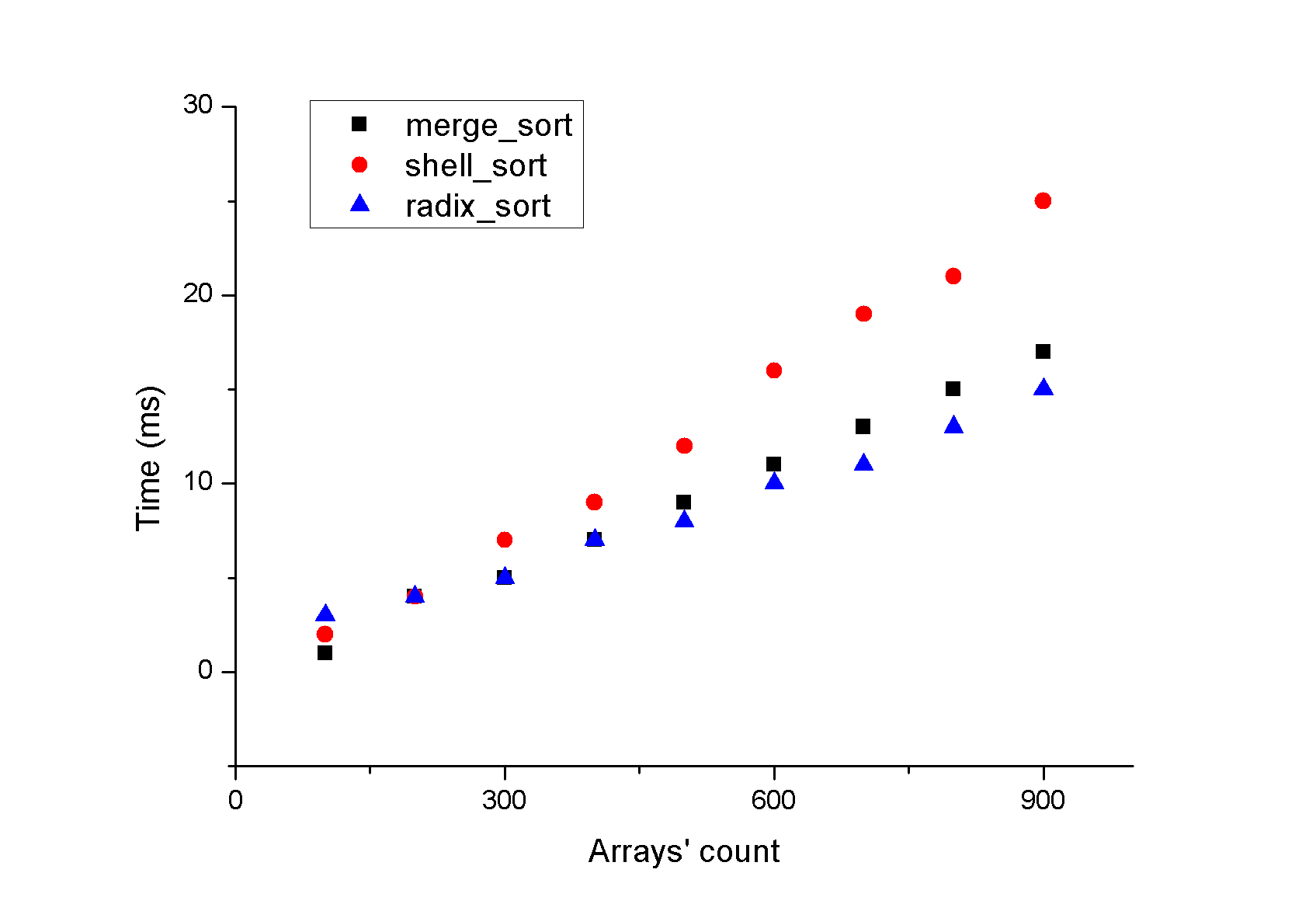
* + 1. Перестановки

*Для поразрядной сортировки невозможно посчитать в связи с реализацией.*

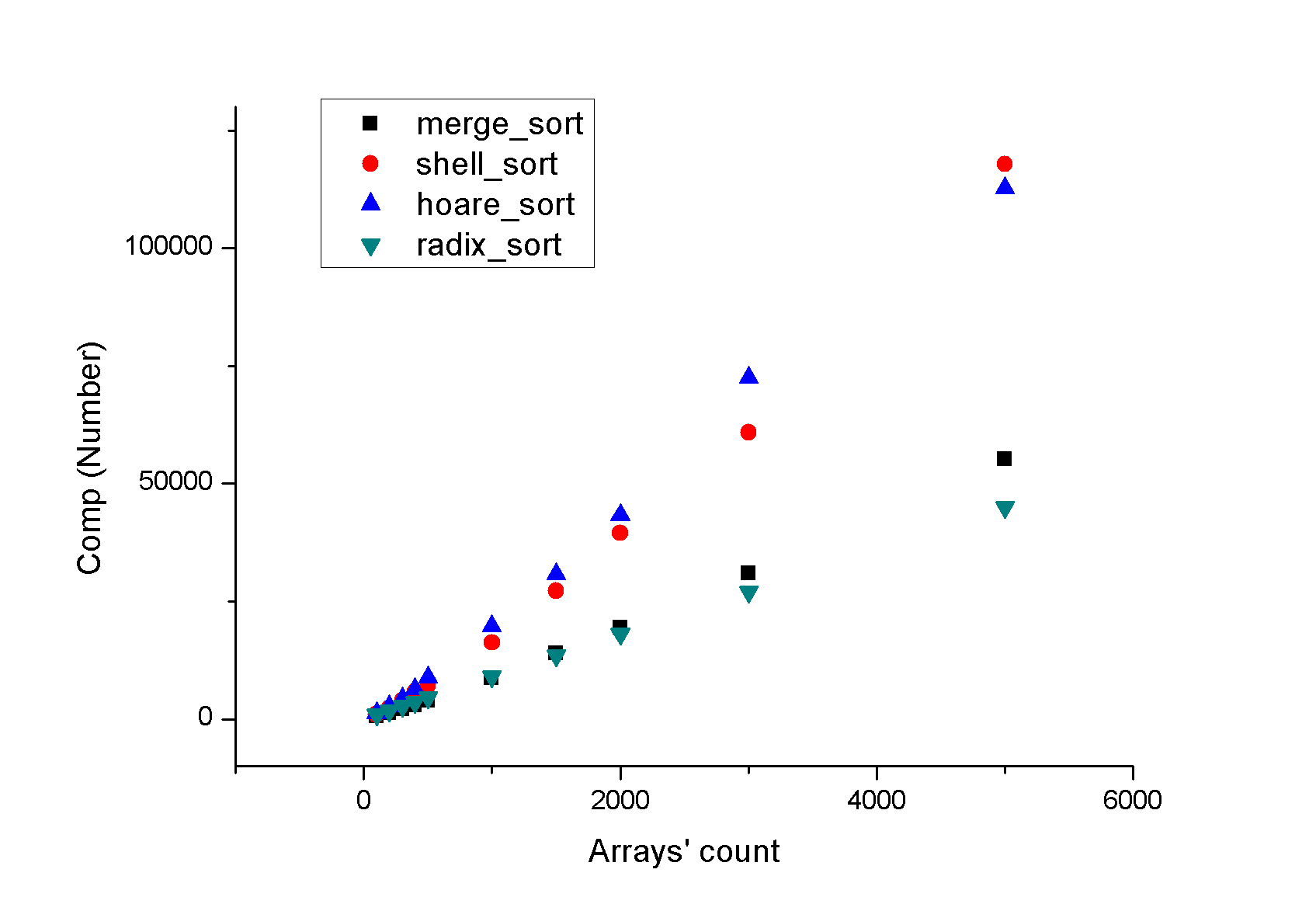


* + 1. Время



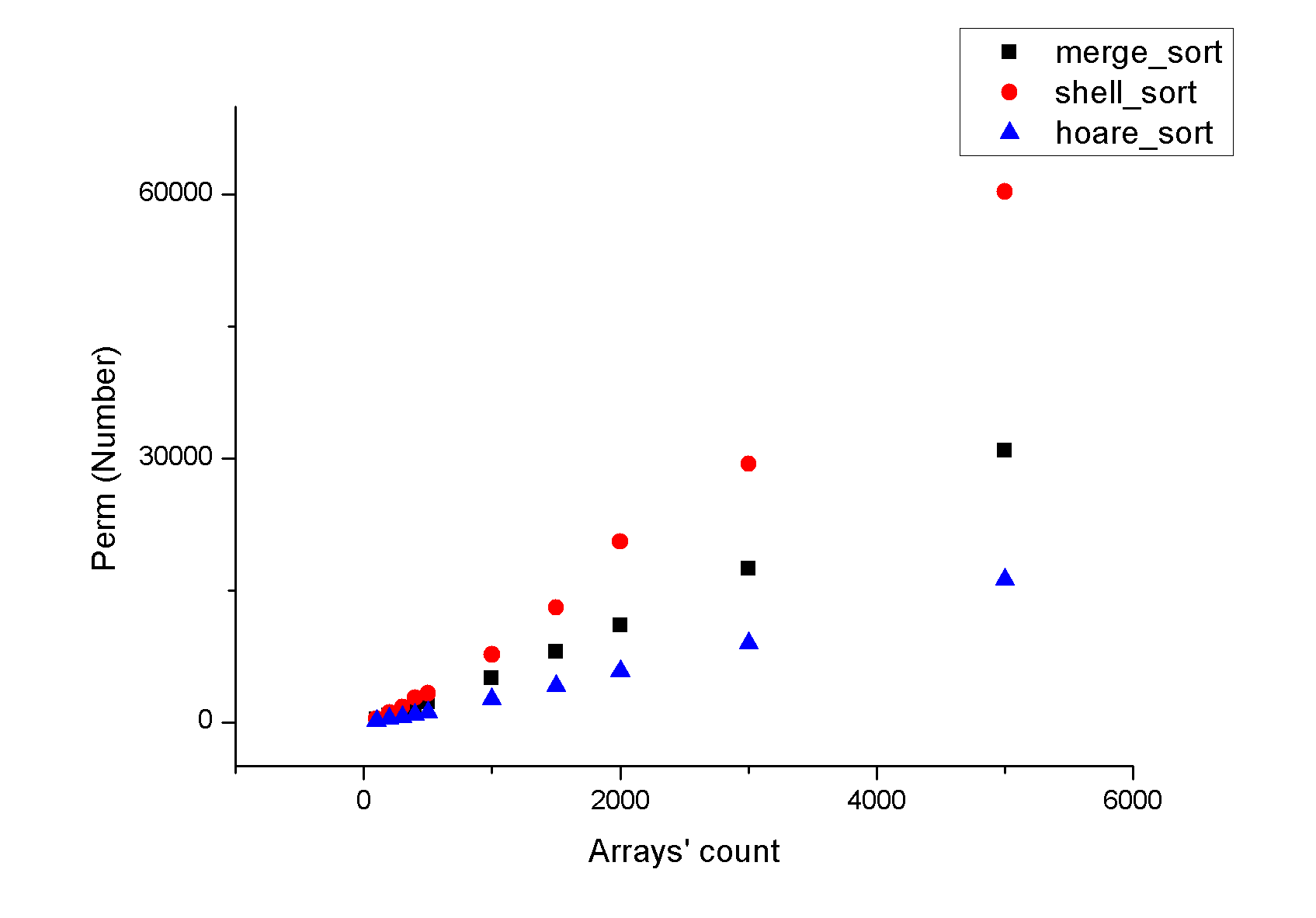


* 1. **Случайные данные:**
     1. Сравнения:

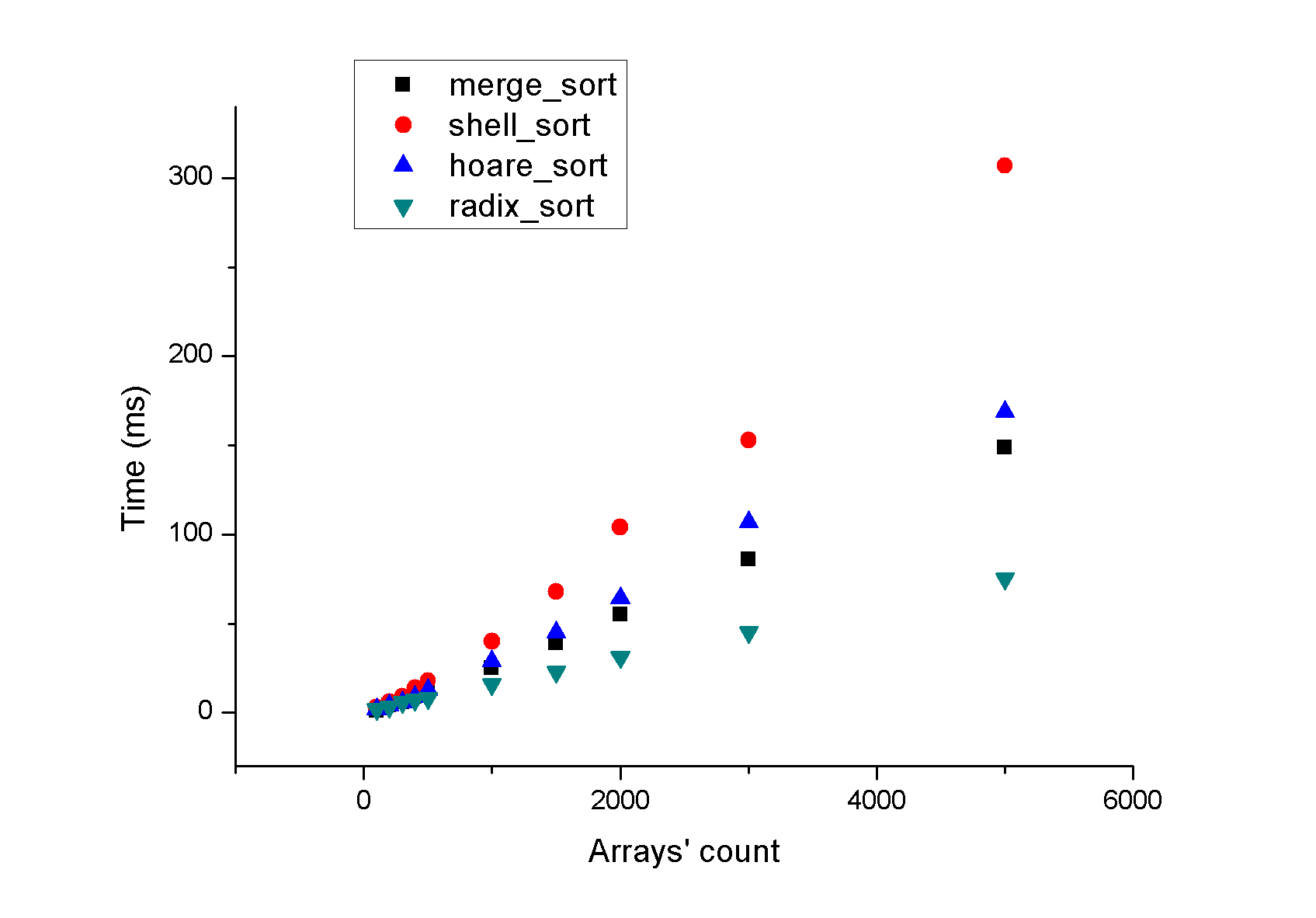


* + 1. Перестановки

*Для поразрядной сортировки невозможно посчитать в связи с реализацией.*



* + 1. Время



1. Обоснование результатов:
   1. Сортировка пузырьком:

В упорядоченном массиве не будет выполнено никаких перестановок, то есть, на них не будет тратиться время. Поэтому, если на каждом шаге цикла данный элемент будет больше предыдущего – они будут меняться местами. Следовательно, худший случай – массив, отсортированный в обратном порядке.

* 1. Сортировка слиянием:

На каждой итерации элементы из временного массива копируются в исходный. То есть, на самом деле, получается не число перестановок, а количество раз, которых в массив что то присвоили.

* 1. Сортировка Шелла:

Так же, как и в сортировке пузырьком, в отсортированном массиве не будет выполнено никаких перестановок, следовательно, это лучший случай.

* 1. Быстрая сортировка:

Если рассматривать сортировку уже отсортированного массива, то наихудший результат по сравнениям можно получить, если использовать в качестве опорного элемента самый правый. Тогда потребуется n-1 операций разделения и общая сложность О(n2*)*. Однако, если каждый раз делить его примерно пополам, то максимальная глубина рекурсии, при которой длина подмассива достигнет 1, составит  log2 *n* , а в результате получится сложность О(nlog2 *n)*

* 1. Поразрядная сортировка:

Наилучший случай – либо массив, состоящий из одной цифры, тогда потребуется 1 итерация внешнего цикла (но тогда и сортировать особо нечего), либо массив, состоящий из разных цифр, имеющих одинаковую разрядность, тогда понадобится минимальное количество итераций внешнего цикла.

1. Анализ результатов:

Сортировка слиянием и поразрядная сортировка показали лучшие временные результаты на всех видах данных.

При условии, что разрядность самого длинного числа в массиве равна k, алгоритм поразрядной сортировки закончит своё выполнение максимум за k пробегов по всему массиву (сложность – O(k\*n)).

При использовании эмпирической последовательности Марцина Циура на отсортированных данных, сортировка Шелла является самой быстрой, однако при сортировке случайных данных время сортировки почти приблизилось к сортировке пузырьком (см. график time\_rnd\_shell\_ciur).

Сортировка слиянием с её сложностью О(n\*log2 n*)* стабильно показывала хороший результат.

Приложение:

* Папка Отчёт содержит все графики, а также данные, на основе которых они были построены.