

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
курса «Анализ алгоритмов»
на тему «Алгоритмы сортировки и расчет эффективности алгоритмов»

Студент: Ильясов И.М., группа ИУ7-53Б

Преподаватели: Волкова Л.Л., Строганов Ю.В.

Москва, 2019 г.

Содержание

Введение	3
1 Аналитическая часть	4
1.1 Описание алгоритмов.....	4
2 Конструкторская часть	5
2.1 IDEF0-диаграмма.....	5
2.2 Разработка алгоритмов	5
2.3 Расчет трудоемкости.....	9
2.3.1 Стандартный алгоритм умножения	9
2.3.2 Сортировка вставками.....	9
2.3.3 Сортировка пузырьком.....	9
3 Технологическая часть	10
3.1 Требования к программному обеспечению.....	10
3.2 Средства реализации	10
3.3 Листинг кода	10
3.4 Вывод	11
4 Экспериментальная часть	12
4.1 Примеры работы	12
4.2 Результаты тестирования	13
4.3 Постановка эксперимента по замеру времени.....	15
4.4 Вывод	17
Заключение.....	18
Список литературы	19

Введение

Целью данной работы является изучение расчета трудоемкости алгоритмов и методов оптимизации алгоритмов на примере трех различных сортировок. Для достижения поставленной цели необходимо решить следующие задачи:

- изучение выбранных алгоритмов сортировок;
- получение навыков расчета трудоемкости алгоритмов;
- получение практических навыков реализации указанных алгоритмов;
- сравнительный анализ худшего, общего и лучшего случаев для каждой сортировки;
- экспериментальное подтверждение различий во временной эффективности алгоритмов сортировки на лучших, общих и худших случаях при помощи разработанного программного обеспечения на материале замеров процессорного времени выполнения реализации на варьирующихся размерах массивов;
- описание и обоснование полученных результатов в отчете о выполненной лабораторной работе, выполненного как расчётно-пояснительная записка к работе.

1 Аналитическая часть

В этом разделе содержатся теоретическое описание алгоритмов и указание области их применения. В данной работе будут рассмотрены следующие алгоритмы:

- Гномья сортировка;
- Сортировка вставками;
- Сортировка пузырьком

1.1 Описание алгоритмов

Гномья сортировка — метод, при котором рассматриваются следующий и предыдущий элементы: если они в правильном порядке, делается шаг на следующий элемент, иначе элементы меняются местами и делается шаг назад. Граничные условия: если нет предыдущего элемента, делается шаг вперед; если нет следующего элемента, конец. Данный метод сортировки имеет оценочную сложность $O(n^2)$, а его оптимизация трансформирует данный алгоритм в сортировку вставками [1].

Сортировка вставками — улучшенный метод гномьей сортировки, в котором элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее среди ранее упорядоченных элементов. Данный метод имеет оценочную сложность $O(n^2)$ [2].

Сортировка пузырьком — один из простейших методов сортировки. Идея метода заключается в следующем: производится обход входной последовательности, по пути просматриваются пары соседних элементов; если элементы некоторой пары находятся в неправильном порядке, то элементы меняются местами. Сложность данного метода равна $O(n^2)$ [3].

2 Конструкторская часть

В данном разделе будут приведены схемы алгоритмов и теоретический расчет их трудоемкости.

2.1 IDEF0-диаграмма

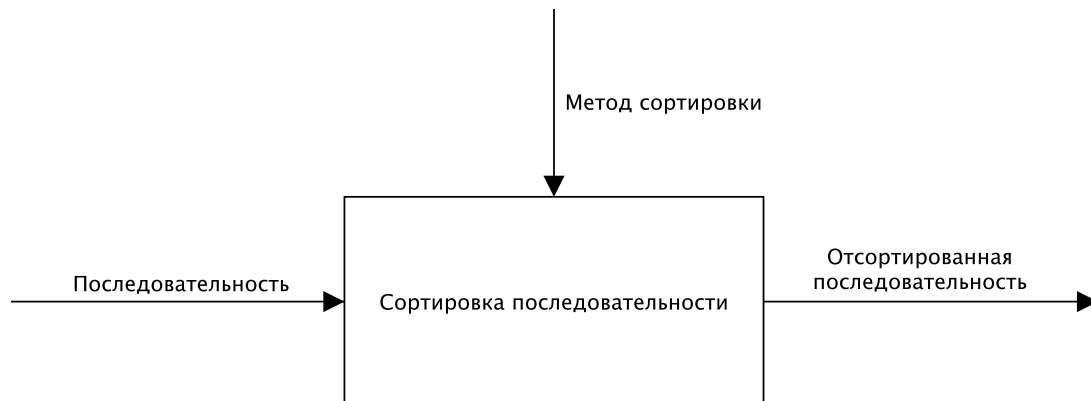


Рисунок 1. IDEF0-диаграмма алгоритма сортировки последовательности

2.2 Разработка алгоритмов

Ниже представлены схемы алгоритмов сортировки последовательностей гномьей сортировкой (рис.2), сортировкой вставками (рис.3) и пузырьком (рис.4).

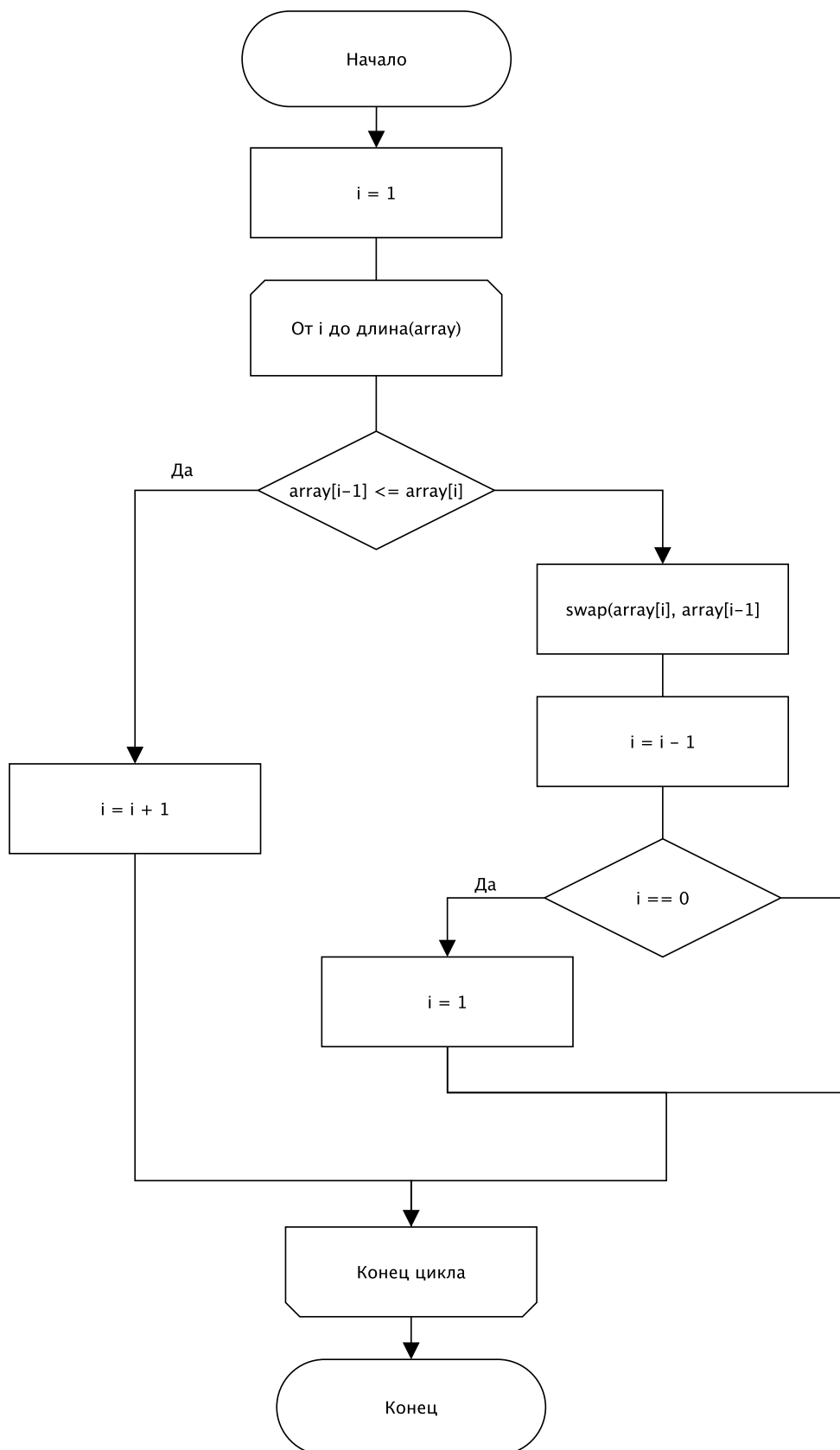


Рисунок 2: схема алгоритма сортировки последовательности гномьей сортировкой

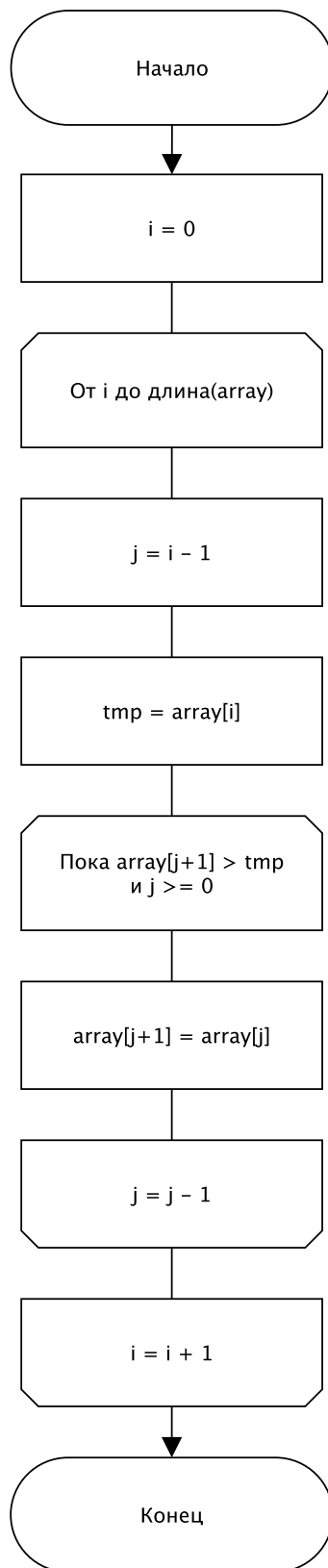


Рисунок 3: схема алгоритма сортировки последовательности сортировкой вставками

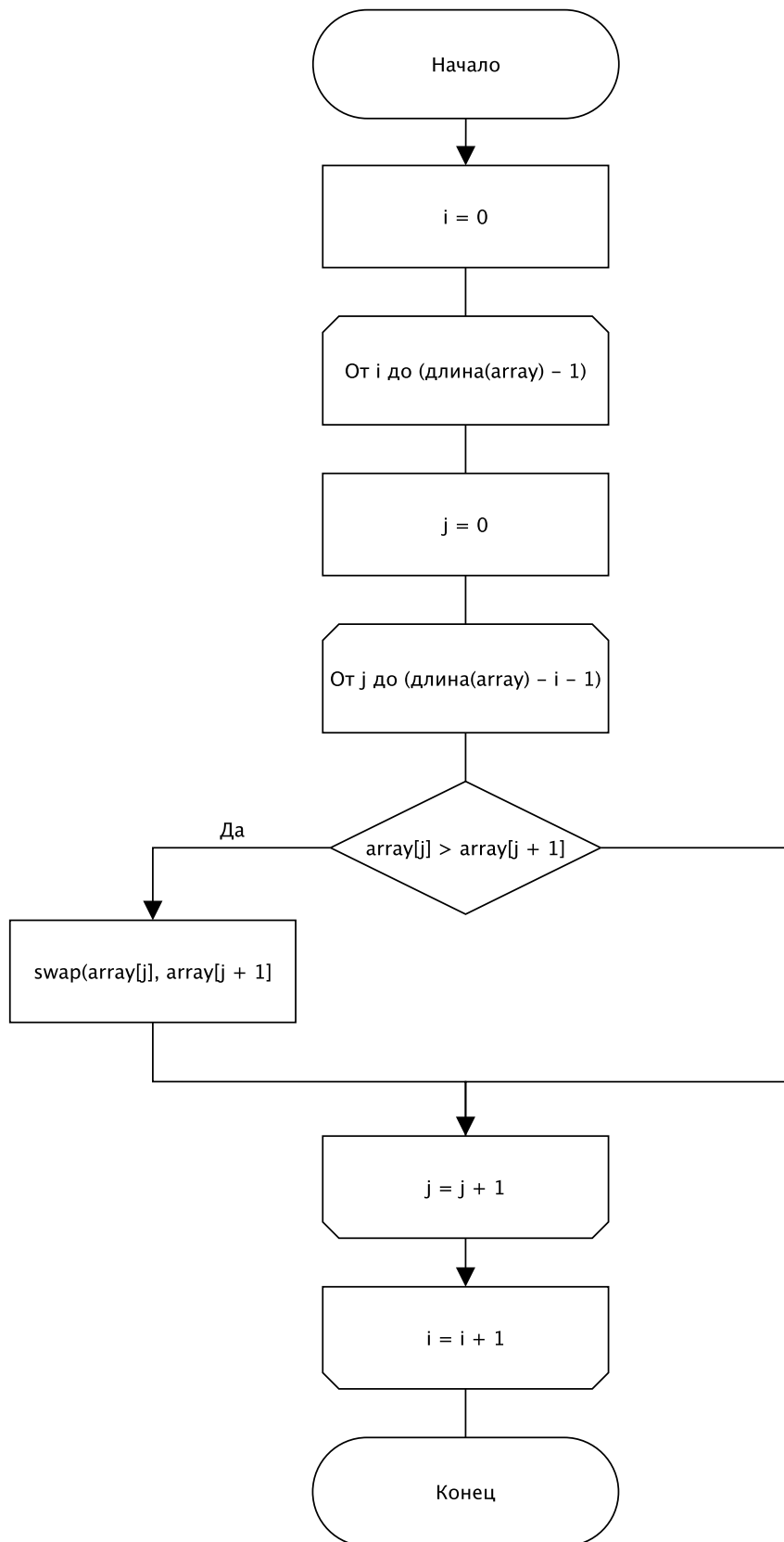


Рисунок 4: схема алгоритма сортировки последовательности сортировкой пузырьком

2.3 Расчет трудоемкости

В этом пункте аналитически выполнен расчет трудоемкости исследуемых алгоритмов. В используемой модели расчета трудоемкости за единицу считаются операции $+$, $-$, $*$, $/$, $=$, $[]$, $>$, $<$, $==$, $<=$, $>=$, $++$, $--$, $+=$, $-=$.

2.3.1 Стандартный алгоритм умножения

Алгоритм гномьей сортировки на псевдокоде выглядит так:

```
while (i < len) {
    if (array[i - 1] <= array[i] || i == 0) {
        i = i + 1
    } else {
        tmp = array[i]
        array[i] = array[i - 1]
        array[i - 1] = tmp
        i = i - 1
    }
}
```

Вычисление трудоёмкости:

$$f = 1 + (7 + \left\{ \frac{1}{8} \right\}) * \left\{ \begin{matrix} N \\ (N + 1) + \frac{2 * 2 + 2 * (N - 1)}{2} * (N - 2) \end{matrix} \right\} = N^2 - 1$$

2.3.2 Сортировка вставками

Согласно источникам, трудоемкость данной сортировки составляет $O(N)$, $O(N^2)$ и $O(N^2)$ в лучшем, худшем и общем случае соответственно [].

2.3.3 Сортировка пузырьком

Согласно источникам, трудоемкость данной сортировки составляет $O(N^2)$ во всех случаях [3].

3 Технологическая часть

В данном разделе представлены требования к разрабатываемому программному обеспечению, средства, использованные в процессе разработки для реализации поставленных задач, а также листинг кода программы.

3.1 Требования к программному обеспечению

Программа должна принимать на вход массив из целых чисел. Числа разделяются переносом строки. Программа должна сортировать массив с помощью трех упомянутых выше алгоритмов — гномьей сортировкой, сортировкой вставками и сортировкой пузырьком. Программа также должна производить замеры времени сортировки на длинах массивов от 100 до 1000 с шагом 100 и строить графики сравнения скоростей лучшего, худшего и общего случаев.

3.2 Средства реализации

Для реализации поставленной задачи был использован язык программирования Python. Для измерения процессорного времени была использован метод `time.process_time()`.

3.3 Листинг кода

В приведенных ниже листингах представлены реализации гномьей сортировки (листинг 1), сортировки вставками (листинг 2) и сортировки пузырьком (листинг 3).

Листинг 1: Гномья сортировка:

```
def gnome_sort(array):
    i = 1
    while i < len(array):
        if (array[i - 1] <= array[i]):
            i = i + 1
        else:
            tmp = array[i]
            array[i] = array[i - 1]
            array[i - 1] = tmp
            i = i - 1
            if (i == 0):
                i = 1
    return array
```

Листинг 2: Сортировка вставками:

```
def insert_sort(array):
    for i in range(len(array)):
        j = i - 1
        tmp = array[i]
        while array[j] > tmp and j >= 0:
            array[j + 1] = array[j]
            j = j - 1
        array[j + 1] = tmp
    return array
```

Листинг 3: Сортировка пузырьком:

```
def bubble_sort(array):
    for i in range(len(array) - 1):
        for j in range(len(array) - i - 1):
            if array[j] > array[j + 1]:
                array[j], array[j + 1] = array[j + 1], array[j]
    return array
```

3.4 Вывод

На основе схем алгоритмов, представленных в конструкторском разделе, в соответствии с указанными требованиями к реализации с использованием средств языка Python было разработано программное обеспечение, содержащее реализации выбранных алгоритмов.

4 Экспериментальная часть

В данном разделе будут приведены примеры работы программы, постановка эксперимента и сравнительный анализ алгоритмов на основе полученных данных.

4.1 Примеры работы

На приведенных ниже рисунках продемонстрирована работа выбранных алгоритмов.

```
Введите длину массива: 4
Введите 0-й элемент массива: 5
Введите 1-й элемент массива: 2
Введите 2-й элемент массива: 0
Введите 3-й элемент массива: 3

Результат сортировки гномьей сортировкой: : 0 2 3 5
Результат сортировки сортировкой вставками: : 0 2 3 5
Результат сортировки сортировкой пузырьком: : 0 2 3 5
```

Рисунок 5: пример №1 работы алгоритмов.

```
Введите длину массива: 7
Введите 0-й элемент массива: 0
Введите 1-й элемент массива: -2
Введите 2-й элемент массива: 5
Введите 3-й элемент массива: 1
Введите 4-й элемент массива: 2
Введите 5-й элемент массива: 7
Введите 6-й элемент массива: 2

Результат сортировки гномьей сортировкой: : -2 0 1 2 2 5 7
Результат сортировки сортировкой вставками: : -2 0 1 2 2 5 7
Результат сортировки сортировкой пузырьком: : -2 0 1 2 2 5 7
```

Рисунок 6: пример №2 работы алгоритмов.

4.2 Результаты тестирования

Таблица 1:

Результаты замеров времени трех реализаций сортировки последовательностей
(лучший случай)

Размерность	Гномья	Вставками	Пузырьком
100x100	0.03840	0.03260	0.76940
200x200	0.05920	0.05320	3.00720
300x300	0.09760	0.08460	6.27140
400x400	0.13280	0.11020	11.54360
500x500	0.16760	0.14420	18.86100
600x600	0.20320	0.17940	28.43720
700x700	0.24500	0.20480	39.76320
800x800	0.27700	0.24380	54.44700
900x900	0.34460	0.28680	72.09480
1000x1000	0.35260	0.32760	88.42020

Таблица 2:

Результаты замеров времени трех реализаций сортировки последовательностей
(худший случай)

Размерность	Гномья	Вставками	Пузырьком
100x100	4.56540	1.26000	1.94080
200x200	17.18460	4.69980	8.08040
300x300	40.87280	10.69200	17.39320
400x400	71.95840	18.90200	29.85800
500x500	114.41540	30.50060	49.69840
600x600	168.19220	44.71780	73.64640
700x700	229.86940	60.63400	100.87740
800x800	305.47040	82.03520	139.30240
900x900	408.53240	107.18120	180.66420
1000x1000	483.50340	129.82600	219.90840

Таблица 3:

Результаты замеров времени трех реализаций сортировки последовательностей
(общий случай)

Размерность	Гномья	Вставками	Пузырьком
100x100	2.44340	0.62020	1.39340
200x200	8.49040	2.51940	5.28180
300x300	20.36680	5.22060	11.86420
400x400	36.73020	9.55880	21.85800
500x500	58.43660	15.23920	34.55020
600x600	85.49420	23.07960	52.03600
700x700	116.34180	30.73880	71.78380
800x800	156.20360	42.51040	97.43000
900x900	206.09080	55.94620	127.70160
1000x1000	250.91080	65.22580	154.82120

4.3 Постановка эксперимента по замеру времени

На рис. 7 представлены результаты замеров для варьирующихся размерностей массива от 100 до 1000 с шагом 100 для всех случаев работы гномьей сортировки, на рис. 8 — для всех случаев работы сортировки вставками, на рис. 9 — для всех случаев работы сортировки пузырьком. Один эксперимент ставился 10 раз, вычислялось среднее время работы.

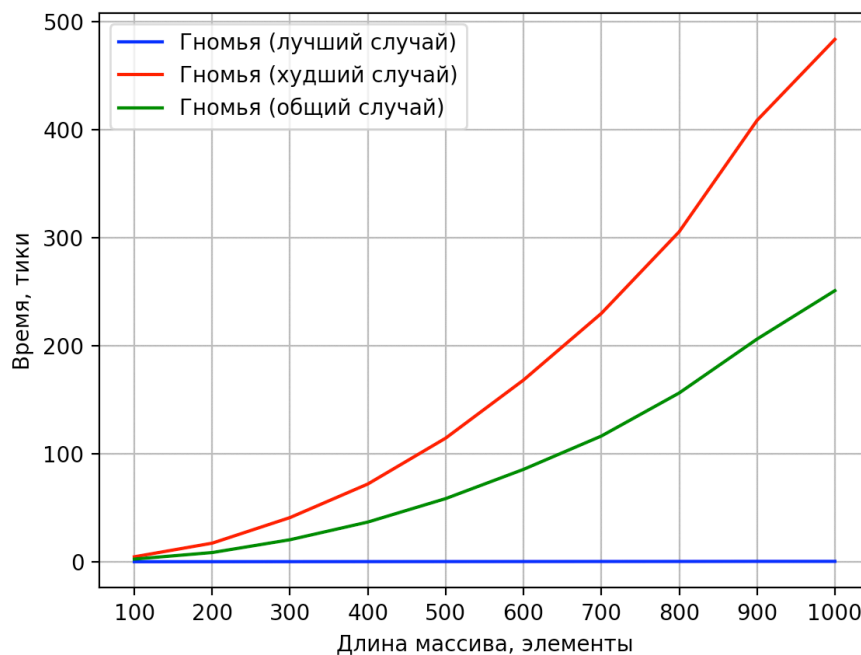


Рисунок 7: график зависимости времени работы гномьей сортировки (лучший, худший и общий случаи) от длины массива.

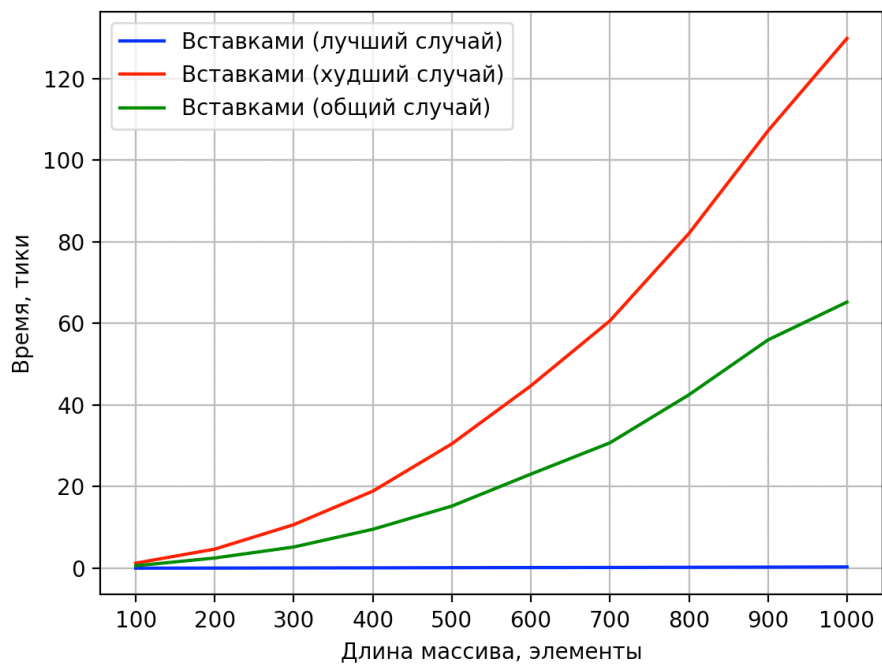


Рисунок 8: график зависимости времени работы сортировки вставками (лучший, худший и общий случаи) от длины массива.

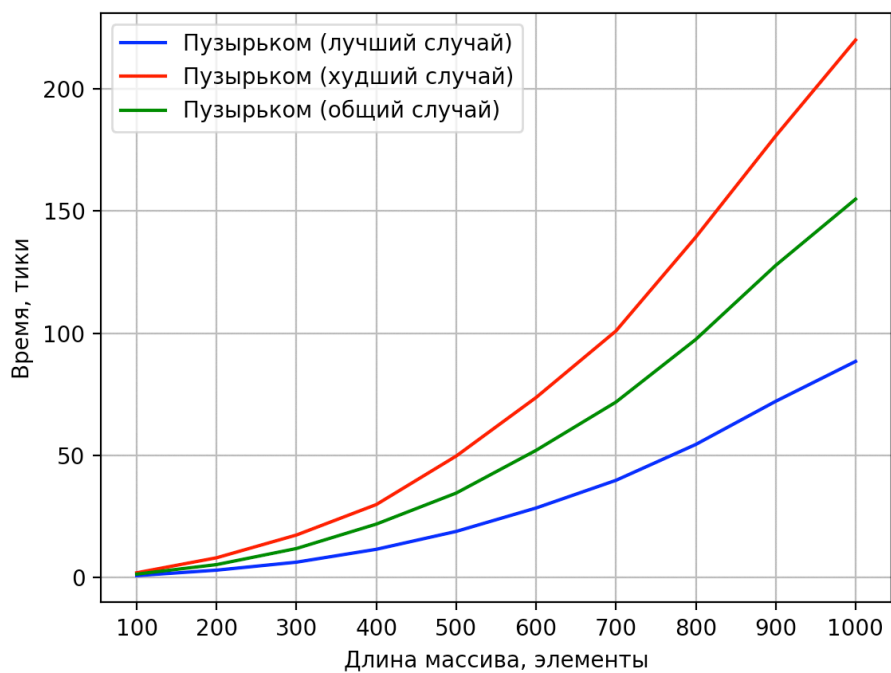


Рисунок 9: график зависимости времени работы сортировки пузырьком (лучший, худший и общий случаи) от длины массива.

4.4 Вывод

По графикам рис. 5, 6, 7 видно, что сортировка вставками эффективнее в худшем случае остальных методов в несколько раз. Это означает, что алгоритм сортировки вставками эффективнее других рассматриваемых алгоритмов. Также можно отметить, что гномья сортировка и сортировка пузырьком в общем случае имеют примерно одинаковую эффективность на длинах массивов до 1000. При этом в лучшем случае эффективность метода вставками и гномьей сортировки идентична.

Заключение

В процессе выполнения лабораторной работы было проведено исследование алгоритмов гномьей сортировки, сортировки вставками и пузырьком массивов различных длин. Во время разработки программного обеспечения в соответствии с поставленными требованиями были получены практические навыки реализации указанных алгоритмов. При помощи разработанного программного обеспечения на материале замеров процессорного времени выполнения реализации на варьирующихся размерах матриц были экспериментально подтверждены различия во временной эффективности гномьей сортировки, сортировки вставками и сортировки пузырьком.

Список литературы

1. "Глупая сортировка и некоторые другие, поумнее // Habr URL: <https://habr.com/ru/post/204968/> (дата обращения: 09.10.2019)"
2. Кнут Д. Э. 5.2 Внутренняя сортировка // Искусство программирования. Том 3. Сортировка и поиск = The Art of Computer Programming. Volume 3. Sorting and Searching / под ред. В. Т. Тертышного (гл. 5) и И. В. Красикова (гл. 6). — 2-е изд. — Москва: Вильямс, 2007. — Т. 3. — 832 с. — ISBN 5-8459-0082-1.
3. Левитин А. В. — "Глава 3. Метод грубой силы: Пузырьковая сортировка // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — С. 144–146. — 576 с. — ISBN 978-5-8459-0987-9"
4. "Пузырьковая сортировка и все-все-все // Habr URL: <https://habr.com/ru/post/204600/> (дата обращения: 09.10.2019)"