

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. М.В. ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики

Отчет по заданию № 1

«Методы сортировки»

Вариант 3 3 2 4

Исполнитель:

Студент гр. 106
Кондрашов Д.С.

Преподаватели:

Корухова Л.С.
Манушин Д.В.



Москва, 2024

Содержание

1	Постановка задачи	1
2	Результаты экспериментов	1
3	Структура программы и спецификация функций	2
4	Отладка программы, тестирование программы	3
5	Анализ допущенных ошибок	3
6	Литература	4

1 Постановка задачи

1. Требуется реализовать два метода сортировки одномерного массива:
 - 1) Сортировка методом простого выбора
 - 2) Быстрая сортировка, основанная на рекурсивной реализации
(Числа упорядочиваются по неубыванию модулей)
2. Сравнить асимптотическую сложность данных алгоритмов.
3. На основе полученных результатов предоставить таблицу сравнений и сделать вывод.

2 Результаты экспериментов

В результате проведенных экспериментов была подтверждена асимптотическая сложность алгоритмов:

Selection sort $O(n \cdot (n - 1)/2)$

Quick sort $O(n \cdot \log_2 n)$

<i>N</i>	Номер сортировки	1	2	3	Среднее значение
10	Сравнения	45,00	45,00	45,00	45,00
	Перемещения	6,82	6,80	6,99	6,87
50	Сравнения	1 225,00	1 225,00	1 225,00	1 225,00
	Перемещения	45,47	45,19	45,67	45,45
100	Сравнения	4 950,00	4 950,00	4 950,00	4 950,00
	Перемещения	94,89	94,62	94,50	94,67
500	Сравнения	124 750,00	124 750,00	124 750,00	124 750,00
	Перемещения	492,40	493,14	493,70	493,08
1000	Сравнения	499 500,00	499 500,00	499 500,00	499 500,00
	Перемещения	992,10	992,40	992,30	992,27
5000	Сравнения	12 497 500,00	12 497 500,00	12 497 500,00	12 497 500,00
	Перемещения	4 990,35	4 990,40	4 991,20	4 990,65
10000	Сравнения	49 995 000,00	49 995 000,00	49 995 000,00	49 995 000,00
	Перемещения	9 989,10	9 989,60	9 990,59	9 989,76

** количество массивов было сокращено с 4 до 3, потому что каждое полученное число - это среднее из 10000 прогонов с различными массивами одной длины

- 1 - элементы упорядочены,
2 - элементы упорядочены в обратном порядке,
3 - расстановка элементов случайна

Вывод:

В результате проведенных опытов были подтверждены формулы, использующиеся для нахождения асимптотической сложности алгоритмов, и было сделано заключение о том, что quick sort быстрее, чем selection sort, но для этого нужны наиболее случайные значения, при росте количества элементов массива эта разница становится заметнее еще больше.

3 Структура программы и спецификация функций

Для более оптимизированной работы программы использовались функции, которые работали как с массивом данных, так и с численными переменными.

Список функций:

1. `change()` – функция меняет местами два элемента массивов, которые ей подаются.
2. `selection_sort()` – функция сортирует массив методом выбора и подсчитывает количество изменений и сравнений, которые были при этом сделаны.
3. `fast_sort()` – функция сортирует массив методом быстрой сортировки рекурсией и подсчитывает количество изменений и сравнений, которые были при этом сделаны.
4. `sort_q()` – сортирует массив по возрастанию, используя алгоритм `selection sort`.
5. `sort_rev()` – сортирует массив в обратном порядке, используя алгоритм `selection sort`.
6. `filling()` – заполняет массив случайными числами, используя условие, которое было выбрано.
7. `sred()` – подсчитывает среднее из всех элементов массива

4 Отладка программы, тестирование программы

Для отладки и соответствующей тестировки программы в каждую из сортирующих функций был добавлен вывод массива после каждого прохода по нему. (Далее будет представлен пример прохода по случайному массиву и его сортировка)

selection_sort

172.00	5992.90	-4900.79	-7914.01	24101.29	5888.81	-12516.97
172.00	-4900.79	5992.90	-7914.01	24101.29	5888.81	-12516.97
172.00	-4900.79	5888.81	-7914.01	24101.29	5992.90	-12516.97
172.00	-4900.79	5888.81	5992.90	24101.29	-7914.01	-12516.97
172.00	-4900.79	5888.81	5992.90	-7914.01	24101.29	-12516.97
172.00	-4900.79	5888.81	5992.90	-7914.01	-12516.97	24101.29

quick_sort

172.00	5992.90	-4900.79	5888.81	24101.29	-7914.01	-12516.97
172.00	-4900.79	5992.90	5888.81	-7914.01	24101.29	-12516.97
172.00	-4900.79	5888.81	5992.90	-7914.01	-12516.97	24101.29

P.S. при проверке отсортированного массива не стоит забывать, что элементы сортируются по неубыванию модулей

5 Анализ допущенных ошибок

При написании функции `fast_sort()` для сортировки массива методом быстрой сортировки (рекурсивная реализация), была допущена ошибка:

значение `pivot` (опорного элемента) сохранялось, как значение индекса этого элемента в массиве, а не как вещественное число, из-за чего при выполнении функции значение `pivot` могло меняться в течение одного прохода, что привело к неправильной сортировке массивов.

6 Литература

Список литературы

- [1] Трифонов Н. П., Пильщиков В. Н. Задания практикума на ЭВМ (1 курс). Методическая разработка(составители). — М.: ВМК МГУ, 2001.
- [2] Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. Второе издание. — М.: «Вильямс», 2005.
- [3] Головинов Г. Основы программирования на TeX. Том 1. Начало. — М.: МФТИ, 2024.
- [4] Кнут Д. Искусство программирования для ЭВМ. Том 3. — М.: Мир, 1978.
- [5] Лорин Г. Сортировка и системы сортировки. — М.: Наука, 1983.