

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ОТЧЕТ ПО ЗАДАНИЮ №1

«Методы сортировки»

Вариант 1 / 1 / 2 / 4

Выполнила:
студентка 104 группы
Соколова А. О.

Преподаватель:
Гуляев Д. А.

Москва
2022

Содержание

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

Постановка задачи

Используя язык программирования С реализовать два метода сортировки: метод простого выбора и быструю сортировку. Провести экспериментальное сравнение их эффективности в решении задачи сортировки по неубыванию. Сравнение должно быть проведено на разных массивах целых чисел (int). При выполнении сортировок на каждом массиве подсчитать число сравнений и число перемещений элементов.

Результаты экспериментов

Сложность выполнения сортировки методом простого выбора в лучшем, среднем и худшем случаях - $O(n^2)$. Количество выполняемых сравнений равно $\frac{n*(n-1)}{2}$, количество выполняемых перемещений равно $n - 1$ [1].

Общее число сравнений быстрой сортировки в лучшем случае равно $n*\log_2 n$, а число перемещений $\frac{n*\log_2 n}{6}$. При средней производительности число сравнений и перемещений умножается на $2 * \ln 2$. В худшем случае производительность метода будет порядка n^2 [1].

Результаты эксперимента приведены ниже в таблицах 1 и 2. В моих результатах общее количество операций, выполняемых методом простого выбора, всегда больше общего количества операций, выполняемых быстрой сортировкой на том же массиве. При этом количество операций перемещения у метода простого выбора всегда не больше, чем у быстрой сортировки. Это соответствует приведенным формулам.

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	9	9	9	9	9
100	Сравнения	4950	4950	4950	4950	4950
	Перемещения	99	99	99	99	99
1000	Сравнения	499500	499500	499500	499500	499500
	Перемещения	999	999	999	999	999
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	9999	9999	9999	9999	9999

Таблица 1: Результаты работы сортировки методом простого выбора

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	19	12	15	12	14
	Перемещения	6	11	10	11	9
100	Сравнения	480	386	441	429	434
	Перемещения	63	112	182	180	134
1000	Сравнения	7987	6996	7557	7809	7587
	Перемещения	511	1010	2597	2631	1687
10000	Сравнения	113631	103644	105709	96732	104929
	Перемещения	5904	10904	33585	33949	21085

Таблица 2: Результаты работы быстрой сортировки

Структура программы и спецификация функций

Функции сортировки:

1. *simple_choice(int n, int *arr, int *count_comp, int *count_move)* – функция, которая сортирует массив методом простого выбора и считает число сравнений и перемещений элементов. Функция не имеет возвращаемых значений и принимает в качестве параметров размер массива, указатель на массив, указатели на переменные, которые отвечают за число сравнений и перемещений.
2. *quick_sort(int n, int *arr, int *count_comp, int *count_move)* – функция, которая сортирует массив методом быстрой сортировки и считает число сравнений и перемещений элементов. Функция не имеет возвращаемых значений и принимает в качестве параметров размер массива, указатель на массив, указатели на переменные, которые отвечают за число сравнений и перемещений.

Функции для генерации массивов:

1. *ascending(int n, int *arr)* – функция, которая заполняет массив размера n числами от 1 до n в порядке возрастания. Функция не имеет возвращаемых значений и принимает в качестве параметров размер массива, указатель на массив.
2. *descending(int n, int *arr)* – функция, которая заполняет массив размера n числами от n до 1 в порядке убывания. Функция не имеет возвращаемых значений и принимает в качестве параметров размер массива, указатель на массив.
3. *random_array(int n, int *arr)* – функция, которая заполняет массив размера n случайными числами, которые генерирует с помощью функции *rand*. Функция не имеет возвращаемых значений и принимает в качестве параметров размер массива, указатель на массив.
4. *generator(int n, int *arr, int par)* – функция, генерирующая массив определенного типа, который определяется в соответствии с переданным параметром. Функция не имеет возвращаемых значений и принимает в качестве параметров размер массива, указатель на массив и параметр, отвечающий за тип массива.

Отладка программы, тестирование функций

Функции сортировки были реализованы в соответствии с описанием в учебных материалах [1], их работа была проверена на трёх самостоятельно выбранных массивах. Другие функции так же были протестированы на трёх различных наборах входных параметров.

Анализ допущенных ошибок

В ходе работы над проектом было допущено три существенные ошибки:

1. Ключ для рандомной генерации передавался в функцию *srand* внутри функции *generator*, из-за этого все рандомно сгенерированные массивы содержали одинаковую последовательность элементов. Данная ошибка была исправлена путём перемещения вызова функции *srand* в тело *main*.
2. Сгенерированный массив не копировался для выполнения второй сортировки на нём, из-за этого быстрая сортировка выполнялась на уже отсортированном массиве. Данная ошибка была исправлена с помощью копирования созданного массива и использования второго указателя при выполнении второго метода сортировки.
3. Перемещение элементов было неправильно расположено в коде функции *simple_choice*. Код *simple_choice* был исправлен в соответствии с алгоритмом, описанным в методических материалах.

Все ошибки были выявлены на этапе получения результатов и сверке их с формулами из учебных материалов.

Список литературы

- [1] Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989