

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. Ломоносова
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ОТЧЕТ ПО ЗАДАНИЮ №1

«Методы сортировки»

Вариант 6

Выполнил:
студент 104 группы
Карнаухов М. А.

Преподаватель:
Гуляев Д. А.

Москва
2026

Содержание

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

Постановка задачи

В данной работе требуется реализовать два метода сортировки массива чисел и провести их экспериментальное сравнение. Сравниваются следующие алгоритмы:

- Insertion Sort (сортировка вставками)
- Merge Sort (сортировка слиянием)

Элементы массивов — целые числа. Сортировка проводится по возрастанию.

Результаты экспериментов

Для оценки эффективности алгоритмов измерялось количество сравнений элементов и количество перемещений для каждого алгоритма. Эксперимент проводился на массивах трёх типов:

- случайный массив,
- отсортированный массив,
- обратно отсортированный массив.

n	Type	Ins. Moves	Ins. Comps	Merge Moves	Merge Comps
10	Random	43	29	68	23
	Sorted	18	9	68	19
	Reverse	63	45	68	15
100	Random	2 483	2 382	1 344	547
	Sorted	198	99	1 344	356
	Reverse	5 148	4 950	1 344	316
1000	Random	252 372	251 370	19 952	8 696
	Sorted	1 998	999	19 952	5 044
	Reverse	501 498	499 500	19 952	4 932
10000	Random	25 011 026	25 001 022	267 232	120 363
	Sorted	19 998	9 999	267 232	69 008
	Reverse	50 014 998	49 995 000	267 232	64 608

Таблица 1: Результаты работы сортировок Insertion Sort и Merge Sort для различных размеров массивов

Анализ результатов

- Insertion Sort демонстрирует квадратичную зависимость операций от размера массива. На случайных и обратно отсортированных массивах количество сравнений и перемещений растет почти пропорционально $O(n^2)$. На отсортированном массиве количество операций минимально, что соответствует лучшему случаю $O(n)$.
- Merge Sort показывает почти одинаковое количество операций на всех типах массивов, что подтверждает теоретическую оценку $O(n \log n)$.
- Сравнение показывает, что Merge Sort эффективнее на больших массивах, а Insertion Sort может быть применим для маленьких или почти отсортированных массивов.

Структура программы и спецификация функций

Программа включает следующие функции:

- `void insertionSort(int n, int a[])` — сортировка массива вставками, подсчет сравнений и перемещений.
- `void mergeSort(int arr[], int left, int right)` — сортировка массива слиянием, подсчет сравнений и перемещений.
- `void merge(int arr[], int left, int mid, int right)` — вспомогательная функция для слияния подмассивов.
- `void generateRandom(int arr[], int n, int maxVal)` — генерация случайного массива.
- `void generateSorted(int arr[], int n)` — генерация уже отсортированного массива.
- `void generateReverse(int arr[], int n)` — генерация обратно отсортированного массива.

Отладка программы, тестирование функций

Тестирование проводилось на трёх типах массивов: случайный, отсортированный и обратно отсортированный. Для каждого теста проверялась корректность сортировки и фиксировались показатели сравнений и перемещений.

Анализ допущенных ошибок

На этапе отладки были выявлены следующие моменты:

- Ошибки при подсчете копирований в функции `merge` — исправлены.
- Некорректная индексация подмассивов при слиянии — исправлена.
- Проверка работы алгоритмов на различных типах массивов позволила убедиться в корректности подсчета операций.