

Московский государственный университет имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики

Отчет по заданию №1

«Сравнительный анализ алгоритмов сортировки»

Выполнил:

студент 391 группы

Ратова Я. Ю.

Преподаватель:

Дмитриев Л. В.

Москва

2025

Цель исследования:

Провести сравнительный анализ времени выполнения трех алгоритмов сортировки: пузырьковой, выбором и быстрой сортировки на различных типах массивов данных.

Условия тестирования

Тестовое окружение: Code::Blocks, компилятор GCC

Инструмент измерения времени: функция `clock()` из `time.h`

Размеры тестовых массивов: от 100 до 50 000 элементов

Типы тестовых данных:

- Случайные массивы
- Уже отсортированные массивы
- Массивы, отсортированные в обратном порядке
- Массивы с зигзагообразным распределением

Результаты тестирования

1. Случайные массивы

На массивах со случайными значениями наблюдается ожидаемая картина производительности:

- Пузырьковая сортировка : показывает наихудшие результаты, время выполнения растет квадратично
- Сортировка выбором: демонстрирует среднюю производительность, лучше пузырьковой, но значительно хуже быстрой
- Быстрая сортировка: показывает наилучшие результаты с временем выполнения $O(n \log n)$

Вывод консоли:

```
Сравнение времени выполнения кодировок:
Размер | Пузырьковая | Выбором | Быстрая
-----|-----|-----|-----
  100 | 0.000000 | 0.000000 | 0.000000
 1000 | 0.001000 | 0.002000 | 0.000000
 5000 | 0.054000 | 0.027000 | 0.001000

Тест с большим массивом (10000 элементов):
Пузырьком: 0.226000 сек, Выбором: 0.103000 сек, Быстрая: 0.001000 сек

Process returned 0 (0x0)   execution time : 0.426 s
Press any key to continue.
```

```
Сравнение времени выполнения кодировок:
Размер | Пузырьковая | Выбором | Быстрая
-----|-----|-----|-----
  100 | 0.000000 | 0.000000 | 0.000000
 10000 | 0.261000 | 0.102000 | 0.001000
 50000 | 7.498000 | 3.057000 | 0.008000

Тест с большим массивом (100000 элементов):
Пузырьком: 46.705000 сек, Выбором: 22.990000 сек, Быстрая: 0.476000 сек

Process returned 0 (0x0)   execution time : 81.142 s
Press any key to continue.
```

2. Уже отсортированные массивы

При работе с предварительно отсортированными данными:

- Пузырьковая сортировка: показывает улучшение производительности благодаря проверке на отсортированность
- Сортировка выбором: сохраняет стабильное время выполнения
- Быстрая сортировка: демонстрирует худшие результаты

Вывод консоли:

```
Сравнение времени выполнения кодировок:
Размер | Пузырьковая | Выбором | Быстрая
-----|-----|-----|-----
  100 | 0.000000 | 0.000000 | 0.000000
 1000 | 0.001000 | 0.001000 | 0.003000
 5000 | 0.029000 | 0.027000 | 0.042000

Тест с большим массивом (10000 элементов):
Пузырьком: 0.108000 сек, Выбором: 0.104000 сек, Быстрая: 0.159000 сек

Process returned 0 (0x0)   execution time : 0.483 s
Press any key to continue.
```

3. Массивы, отсортированные в обратном порядке

Наихудший случай для большинства алгоритмов:

- Пузырьковая сортировка: крайне низкая производительность
- Сортировка выбором: результаты лучше предыдущих, но всё равно низкие
- Быстрая сортировка: работает хуже сортировки выбором

Вывод консоли:

```
Сравнение времени выполнения кодировок:
Размер | Пузырьковая | Выбором | Быстрая
-----|-----|-----|-----
100    | 0.000000    | 0.000000 | 0.000000
1000   | 0.003000    | 0.001000 | 0.003000
5000   | 0.048000    | 0.025000 | 0.035000

Тест с большим массивом (10000 элементов):
Пузырьком: 0.185000 сек, Выбором: 0.098000 сек, Быстрая: 0.127000 сек

Process returned 0 (0x0)   execution time : 0.535 s
Press any key to continue.
```

При достижении 50 000 элементов пузырьковая сортировка и сортировка выбором становятся практически неприменимыми из-за чрезмерного времени выполнения.

Быстрая сортировка продолжает демонстрировать приемлемую производительность

4. Массивы, отсортированные зигзагообразно

- Пузырьковая сортировка: показывает улучшение производительности в сравнении с предыдущем пунктом
- Сортировка выбором: результаты хуже всего
- Быстрая сортировка: сохраняет относительную эффективность

Вывод консоли:

```
Сравнение времени выполнения кодировок:
Размер | Пузырьковая | Выбором | Быстрая
-----|-----|-----|-----
100    | 0.000000    | 0.000000 | 0.000000
1000   | 0.003000    | 0.000000 | 0.000000
5000   | 0.040000    | 0.028000 | 0.004000

Тест с большим массивом (10000 элементов):
Пузырьком: 0.142000 сек, Выбором: 0.124000 сек, Быстрая: 0.034000 сек

Process returned 0 (0x0)   execution time : 0.384 s
Press any key to continue.
```

Выводы и рекомендации

1. Пузырьковая сортировка:
 - Подходит только для небольших массивов (до 1000 элементов)
2. Сортировка выбором:
 - Приемлема для малых и средних массивов
 - Простая реализация, но низкая производительность на больших данных
3. Быстрая сортировка:
 - Эффективна на массивах любого размера и типа
 - Уступает по эффективности в уже отсортированных массивах

Заключение

Для промышленного применения настоятельно рекомендуется использование быстрой сортировки или других алгоритмов класса $O(n \log n)$. Пузырьковая сортировка и сортировка выбором могут использоваться только для очень малых объемов данных.