

Организационные вопросы

Оценка лабораторных работ

Лабораторная работа оценивается по 2-балльной шкале (зачтено/не зачтено). Для получения положительной оценки достаточно выполнить работу с соблюдением минимальных требований. Выполнение дополнительных заданий не отменяет необходимость соблюдения минимальных требований.

Минимальные требования

1. Код должен компилироваться без предупреждений при максимальном уровне предупреждений. Для компилятора GCC это набор флагов: `-Wall -Wconversion -Wextra -Wpedantic`. Для компилятора MSVC (Visual Studio) это флаг `/W4`. Для других компиляторов согласуйте настройки компиляции с преподавателем.

2. Нельзя использовать глобальные переменные (константы допустимы).

3. В коде должен использоваться только полноценный английский язык (транслит запрещен). Русский язык разрешено использовать только в комментариях.

4. Запрещается комментировать каждую строчку кода. Допустим один краткий комментарий на блок кода. Разрешается комментировать одиночную строчку кода, только если она действительно делает что-то неожиданное и хитрое (но помните, хитрый код — плохой код!).

5. Весь код должен удовлетворять единому стилю программирования. Сам стиль можно выбирать по своему вкусу (см., например, [Google C++ Style Guide](#), [GNU C Style](#) и другие). То есть запрещено, например, называть одну функцию в стиле [CamelCase](#), а другую — в стиле [snake_case](#). Исключение допустимо только для названия функции `main`, которое всегда пишется в нижнем регистре. Данные требования предъявляются к любым именованным сущностям в программе — к функциям, методам, классам, локальным переменным, параметрам функций и методов, названиям файлов и так далее.

6. В случае удалённого формата сдачи обучающийся предоставляет ссылку на git-репозиторий. В случае очного формата сдачи репозиторий может быть локальным (на компьютере в аудитории или ноутбуке обучающегося). В любом случае репозиторий не может содержать одинокий коммит с целой лабой, а должен показывать историю работы над лабораторной в виде серии коммитов с содержательными заголовками. В репозитории должен быть корректным образом настроен файл `.gitignore` (т. е. в репозитории должны находиться только файлы с исходным кодом и файлами проекта, никаких промежуточных и итоговых результатов компиляции в нём быть не должно).

7. В случае удалённого формата сдачи для защиты необходимо предоставить небольшой содержательный отчёт, подготовленный в Latex, LibreOffice, Microsoft Word или любой другой системе. Отчёт предоставляется в формате pdf (но быть готовым предоставить в исходном формате по просьбе преподавателя). В отчёте должны присутствовать: титульный лист, выданное согласно варианту задание, протокол тестирования, заключение по выполненной работе и дополнительные разделы, требуемые вашим преподавателем (если есть).

Лабораторная работа № 2: Сортировки

Лабораторная работа состоит из 2 заданий:

Задание 1.

Реализовать алгоритмы сортировки для массивов целых чисел согласно своему варианту. Реализуемые сортировки определяются цифрами представления вашего номера варианта в системе счисления по основанию 3.

Группа 1	Группа 2	Группа 3 (по желанию)
0. Сортировка пузырьком 1. Сортировка вставками 2. Сортировка выбором	0. Сортировка Шелла 1. Шейкерная сортировка 2. Быстрая сортировка	0. Сортировка расчёской 1. Сортировка естественным двухпутевым слиянием (natural two-way merge sort) 2. Пирамидальная сортировка

Пример. Допустимый набор вариантов 1—26 (001_3 — 222_3). Предположим, что вам назначен вариант 19. Представим это число в системе счисления с основанием 3:
 $19 = 2 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0 = 201_3$. Таким образом, в 19-ом варианте необходимо реализовать на минимальном уровне сложности:

- сортировку выбором (2);
- сортировку Шелла (0);
- (по желанию) сортировку естественным двухпутевым слиянием (1).

Задание 2.

Посчитать число сравнений и число копирований объектов для массивов длины 1 000, 2 000, 3 000, ..., 10 000, 25 000, 50 000, 100 000:

- а) в среднем (сгенерировать 100 случайных массивов и посчитать средние значения);
- б) для полностью отсортированного массива;
- в) для обратно отсортированного массива.

По полученным данным построить графики, сравнить экспериментальные результаты с теорией, сделать выводы.

Минимальные требования:

1. Запрещено использование функций из заголовочных файлов `<algorithm>` и `<numeric>`.
2. Функции сортировки должны возвращать структуры stats:

```
struct stats {  
    size_t comparison_count = 0;
```

```
size_t copy_count = 0;  
};
```

3. Функции сортировки должны принимать `std::vector<int>` - сортируемый набор элементов.

Дополнительные задания:

0. Реализуйте сортировку из 3 группы.
1. Дополнительно считайте время сортировок (см. `<chrono>`).
2. Измените функции так, чтобы они принимали в себя 2 итератора, указывающих на начало сортируемого диапазона, и конец сортируемого диапазона (подобно функциям из `<algorithm>`).
3. Сделайте ваши функции шаблонными. Продемонстрируйте работоспособность для `std::string` и вашего собственного класса с перегруженным оператором сравнения.