

Знакомство с алгоритмами

(заполняется спикером урока)

Чек-лист:

1. Код компилируется и запускается.
2. Время работы алгоритмов адекватное (соответствует сложности алгоритмов). Например, сортировка за $O(n^2)$ должна работать в разы медленнее сортировки за $O(n \log n)$.
3. В коде корректно замеряется время работы алгоритмов.
4. Алгоритмы сортировки работают корректно - для массива из случайных чисел на выходе всегда получаем отсортированный массив.

Типовые ошибки при выполнении задания:

- Замеры времени включают вывод содержимого массивов в консоль и заполнение данных в массив.
- Массив состоит не из случайных чисел.
- Размер массива маленький (не соответствует указанному в задании).
- Массив не перезаполняется новыми числами после предыдущего алгоритма сортировки.

Само домашнее задание из примера (для вашего понимания):

1. Реализуй два алгоритма сортировки (можно реализовать рассмотренные на занятии) и сравни их скорость работы со стандартным алгоритмом `Arrays.sort()`, реализованным в Java.
2. Сделай замеры скорости работы всех использованных в работе алгоритмов.
3. Сделай выводы: какой из алгоритмов работает быстрее, насколько по времени и какой из этих алгоритмов ты будешь использовать в дальнейшем и почему?

Для проверки времени сортировки лучше брать большие объемы данных – 1000, 10_000 и 100_000 элементов в массиве (лучше случайных, чтобы не было зависимости на конкретные данные в массиве)

Для проверки домашнего задания вышли на проверку свои выводы в текстовом формате, а также ссылку на репозиторий с кодом (реализованные алгоритмы + замеры времени работы алгоритмов)