



ВЫСШАЯ ШКОЛА ЭКОНОМИКИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Департамент программной инженерии  
Алгоритмы и структуры данных - 2

# Экспериментальное сравнение сортировок

2020-2021

# Цель работы

**Цель работы** – экспериментальное определение временной сложности сортировок различных массивов разными методами и сравнительный анализ полученных результатов

<https://www.youtube.com/watch?v=es2T6KY45cA&t=217s> MergeSort vs QuickSort

<https://www.youtube.com/watch?v=kPRA0W1kECg> 15 сортировок

<https://www.youtube.com/watch?v=DSMCZZGbZo4> 50 сортировок

# Задание

Провести эксперименты – измерить **время выполнения** алгоритмов сортировки

- 1) выбором
- 2) пузырьком
- 3) пузырьком с условием Айверсона 1
- 4) пузырьком с условием Айверсона 1+2
- 5) простыми вставками
- 6) бинарными вставками
- 7) подсчетом (устойчивая)
- 8) цифровой
- 9) слиянием
- 10) быстрой (любой опорный, на ваш выбор)
- 11) пирамидальной

Можно выбрать три любые итерационные сортировки ( 1- 6)

Линейные сортировки обе обязательны (7, 8)

Рекурсивные сортировки все три обязательны (9 – 11)

# Задание (продолжение)

1. Измерения провести для размеров массива
  - от 50 до 300, шаг 50
  - от 100 до 4100, шаг 1000
2. Измерения выполнить для массивов, заполненных целыми неотрицательными числами:
  - случайными значениями от 0 до 5
  - случайными значениями от 0 до 4000
  - «Почти» отсортированными в требуемом порядке числами (например, в каждой тысяче элементов поменять местами N пар элементов отсортированного массива)
  - Отсортированными в обратном порядке (по убыванию) числами от 4100 до 1
3. Измерения провести для сортировок 8-ю разными методами
4. Результаты измерений времени сортировки разных массивов разных размеров разными методами вывести в файл / файлы \*.csv (разделители – точки с запятой «;» )

# Задание. Измерение времени

ВАЖНО!!!

## ***Про количество замеров и усреднение времени.***

При работе программы одновременно выполняется большое количество других программ, что влияет на измеряемое время. Как минимизировать это влияние?

- Во время выполнения эксперимента отключите все остальные программы, какие возможно, т.к. они сильно влияют на измеряемое время. В частности, отключитесь от сети.
- Измерять время каждой сортировки надо несколько раз (10, 50, 100 – на ваш выбор), затем время усреднить.

## ***Про оптимизирующий компилятор***

- Первые прогоны программы выполняются медленнее, т.к. оптимизирующий компилятор еще не провел оптимизацию.
- Лучше исключить 2-3 первых замера времени сортировок из расчета среднего времени
- ДО измерений времени прогоните все сортировки «вхолостую»

# Задание. Измерение времени

Измерять время надо обязательно подходящими **инструментами**.

Время сортировки не должно быть равно 0.

Минимальная единица измерения времени в компьютере – 1 такт =  $1/(\text{частота CPU})$  сек.

Далее – наносекунды ( $10^{-9}$  с.), микросекунды ( $10^{-6}$  с.), миллисекунды ( $10^{-3}$  с.), секунды и т.д.

Таймеры измеряют время с точностью до 15,6 мс. Они не годятся для измерения времени в нашем эксперименте.

Надо снимать показания со счетчика тактов.

Можно использовать

- функцию chrono, см. материал, подготовленный студентом гр. БПИ173 Кириллом Быковым <https://darkydash.github.io/cpp-chrono/>
- функции windows измерения частоты и количества тактов CPU, (см. презентацию)
- ассемблерные вставки, в которых также снимается количество тактов (см. презентацию по измерению времени)

# Замечание о сортируемом массиве

Сортировать разными методами надо один и тот же массив.

Не надо сортировать сортированный на предыдущем шаге!!!

В цикле для подсчета количества операций сгенерировать эталонный массив максимальной длины 4100 (4 раза – для 4-х видов массивов), затем перед каждой сортировкой копировать элементы эталонного массива в рабочий массив, который и сортировать.

Сначала копируете 50 элементов, потом 100 и т.д. для первого диапазона размерностей массивов

Сначала копируете 100 элементов, потом 1100 и т.д. для второго диапазона размерностей массивов

Всего сортировок  $(4 \text{ вида массива} * (6 + 5) \text{ длин массива} * 8 \text{ методов}) = 352$

При многократном измерении времени с последующем усреднением количество измерений(и выполнения сортировок) увеличивается. Например, если измерения проводятся по 10 раз для каждого набора изменяемых аргументов, получим 3520 измерений

# План эксперимента

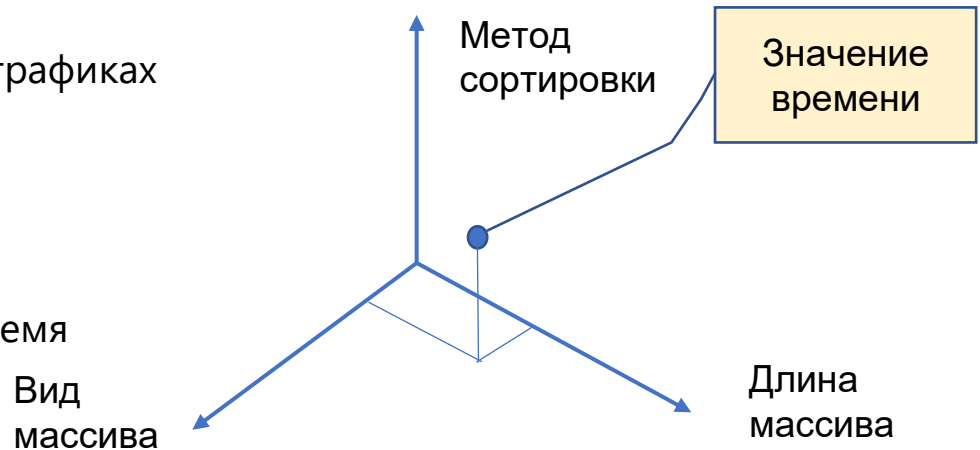
Ищем зависимость времени сортировки от длины массива, изменяя

- Длину массива
- Вид массива
- Метод сортировки

Результаты можно показать на плоских графиках по разрезам пространства измерений.

На всех графиках

- Ось X – размерность массива
- Ось Y – измеренное среднее время





# Задание. Результаты

Представить результаты экспериментов в виде таблиц и графиков (в ЭТ)

«размер массива (ось x) – время сортировки (ось y), в нс или в мкс».

1. 4 + 4 графика, для каждого вида массива. Заголовок – вид массива (случайный 0-5, случайный 0-4000 и т.п.). Легенды – метод сортировки (4 графика, т.к 4 вида заполнения массива, на каждом по 8 кривых, соответствующих восьми методам сортировки)
2. 8 + 8 графиков, для каждого метода сортировки. Заголовок графика – метод сортировки. Легенды – вид массива (на каждом графике по 4 кривых, по количеству разных массивов)

Обязательно проанализировать полученные результаты и сделать выводы – в отдельном текстовом файле

# Таблицы результатов

Представить результаты экспериментов в виде таблиц на листах в Excel, в которых указать: Метод сортировки, вид массива (например, «пузырек, случайные [0;5]» или «Простые вставки, обратно сортированный»)

**Примерный** вид таблиц (должно быть 2 таблицы для двух диапазонов размерностей):

Размер массива	Пузырек Случайные числа 0-5	Пузырек случ Случайные до макс	...	...	Цифровая обратно сортир	Цифровая почти сортир
50						
...						
300						

Для этого выводите результаты измерений времени в файлы \*.csv с разделителем «;» или другим.

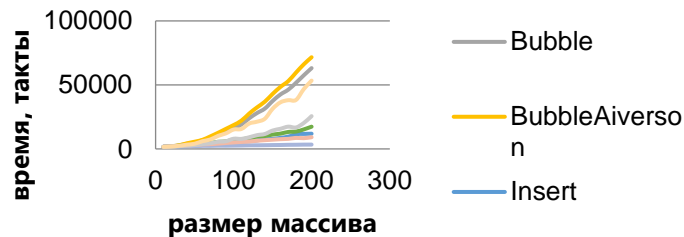
Потом загрузите эти файлы в ЭТ (данные - импорт)

Размер таблиц:  $(6 \text{ строк или столбцов}) * (8 * 4 \text{ столбцов или строк}) = 6 * 32 \text{ (или } 32 * 6)$

$(5 \text{ строк или столбцов}) * (8 * 4 \text{ столбцов или строк}) = 5 * 32 \text{ (или } 32 * 5)$

# Примеры графиков 1

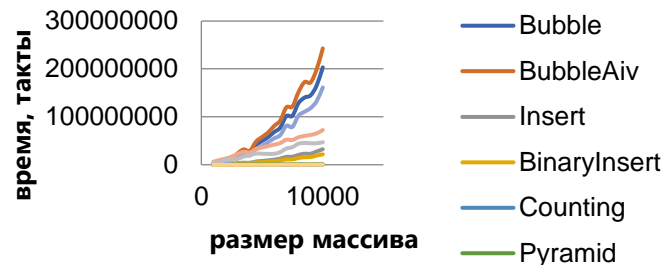
## значения массива - случайные 0-5



Оси на графиках должны быть оцифрованы и обозначены, обязательно укажите единицы измерения времени

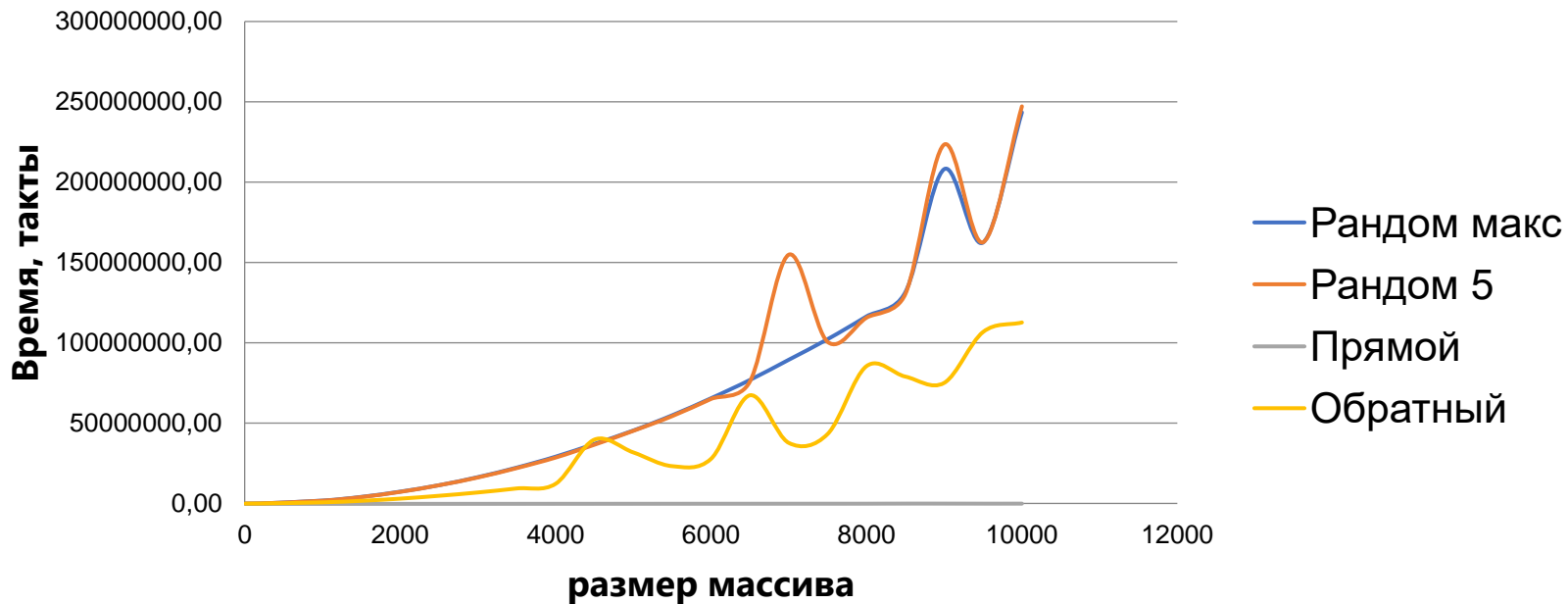
Если при размещении всех методов на одном графике для одного вида исходного массива невозможно их сравнить (некоторые графики «лежат» на оси X), можно разделить методы на две группы и построить два графика)

## обратно сортированный массив



# Пример графиков 2

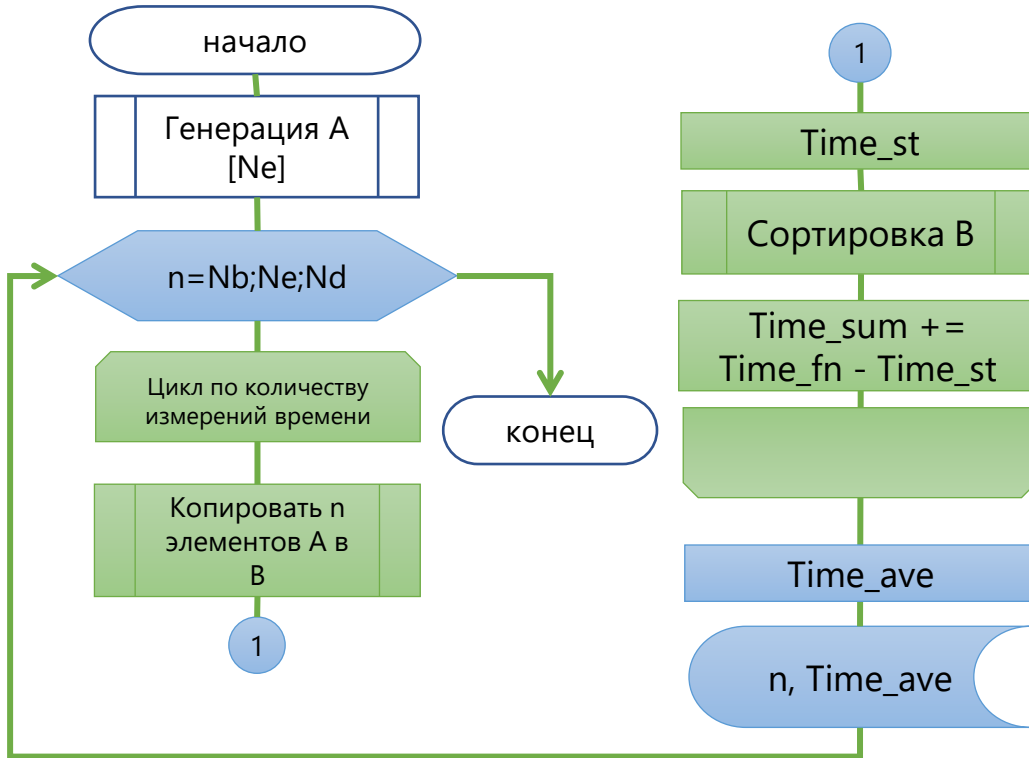
## Пузырек с условием Айверсона 1



# Эксперимент попроще (один вид массива, один метод сортировки)

Nb = 50;  
Ne = 300;  
Nd = 50;

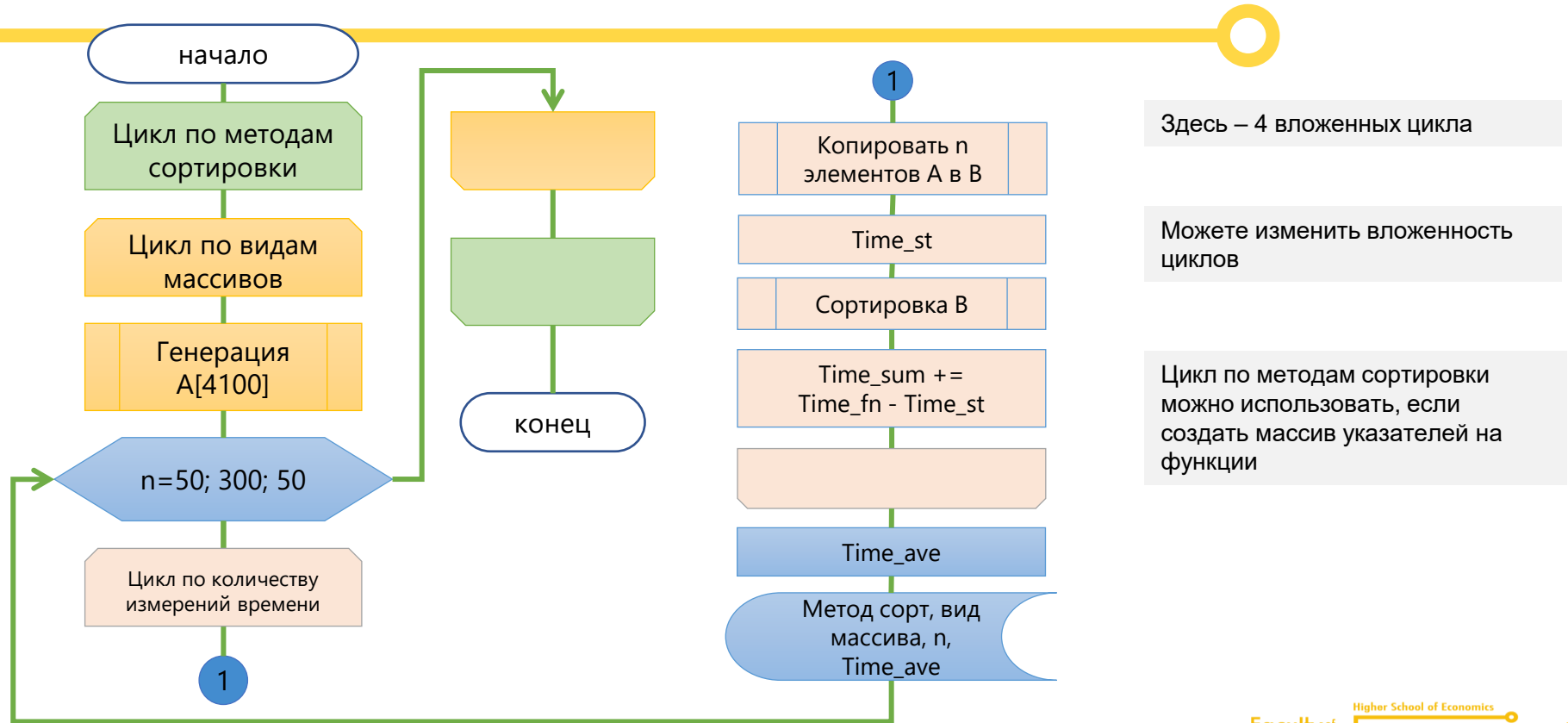
Nb = 100;  
Ne = 4100;  
Nd = 1000;



Здесь два цикла

- по количеству элементов массива n
- и
- по количеству измерений времени

# Эксперимент сложнее (разные виды массива и методы сортировки)



# Обозначения в блок-схемах

A – эталонный массив

B – рабочий массив

Time\_\* - время в нс или мкс

n – текущий размер массива для сортировки

Nb – начальный размер массива

Ne – конечный размер массива

Nd – шаг изменения размера массива

# Проверки

Напишите функцию проверки упорядоченности элементов массива. Она должна возвращать

Истину или 1, если массив сортирован,  
Ложь или 0 – если массив не сортирован.

Предусмотрите вывод исходного и сортированного массивов в файлы input.txt и output.txt (для защиты проекта)



# Комментарии в коде



В начале кода обязательно укажите

- 1) АиСД-2, 2021, задание 5
- 2) ФИО и группа
- 3) Среда разработки
- 4) Перечислите, что сделано
- 5) Перечислите, что не сделано

Далее в коде пишите комментарии, не слишком много, но чтобы было понятно, что в этом фрагменте (цикле, вводе и проч.) выполняется

# Загрузка результатов в ЛМС

**В ЛМС загружаем** один или несколько архивов \*.zip, содержащих

- Комментированный код (только файлы \*.h и \*.cpp)
- Результаты эксперимента в электротаблице - таблицы и графики.
- Текст с анализом полученных результатов и выводами
- Имена файлов таблиц и текста - <группа>\_<FIO>\*.\*

Критерии проверки и оценивания проекта см. в таблице Ведомость



# Возможные ошибки и недостатки

- нет рандомной генерации или не меняется seed (не используется srand);
- неправильно генерируется массив какого-то типа;
- изменяется эталонный массив или повторно подаётся на вход алгоритму отсортированный на предыдущем шаге массив;
- неправильно работают или не реализованы алгоритмы сортировки;
- наличие в функциях сортировок посторонних операций (копирование массивов, вывод и т.п.);
- используется дополнительная память там, где это не требуется;
- неудачно выбран инструмент измерения времени (например, время сортировки = 0),
- измерения для алгоритма отсутствуют (без уважительной причины) или выполняются однократно;
- не создаётся CSV-файл с результатами измерений для дальнейшего создания и обработки таблиц;
- нет таблиц и графиков;
- графики есть, но они «слепые»: без названия, без обозначений осей и/или без легенд;
- анализ полученных результатов и выводы не полные или отсутствуют;
- нет очистки памяти;
- нет комментариев в коде.

# Возможные достоинства работы

- + все минусы предыдущего слайда можно обратить в плюсы, если сделать все верно;
- + использование указателей на функции в качестве параметров функций;
- + полный и качественный анализ результатов;
- + наличие всех графиков, правильно оформленных;
- + отличные ответы на защите;
- + досрочная защита проекта (по договоренности с семинаристом)

# Консультации и защиты задания

Консультации как обычно – в discord, также можно писать в тг или на почту

Защита очная на семинарах по расписанию, преподавателю – семинаристу вашей группы.

Если не хватит времени, назначим дополнительное время.

Будет ссылка на таблицу для записи на защиту (по 5-7 минут)

На защите будут вопросы по

- коду,
- алгоритмам,
- организации экспериментов,
- результатам экспериментов (таблицы и графики)
- анализу и выводам по полученным результатам.

