

# Полученные результаты

## Результаты для MERGE SORT:

На основе эмпирических данных алгоритм подтверждает теоретическую сложность  $O(n \log n)$ . MERGE SORT показывает высокую эффективность при сортировке больших массивов. Алгоритм устойчив к структуре входных данных (случайные, отсортированные или наоборот отсортированные массивы). Однако, на маленьких массивах его рекурсивная структура и использование временных массивов создают дополнительную нагрузку, что приводит к сравнительно большим затратам.

## Результаты для MERGE+INSERTION SORT:

На массивах небольшого размера (до некоторого порогового значения) гибридный алгоритм работает быстрее, так как INSERTION SORT имеет меньшие константные затраты. При превышении порога наблюдается низкая эффективность, так как квадратичная сложность INSERTION SORT начинает преобладать над сложностью MERGE SORT.

### Пороговое значение k:

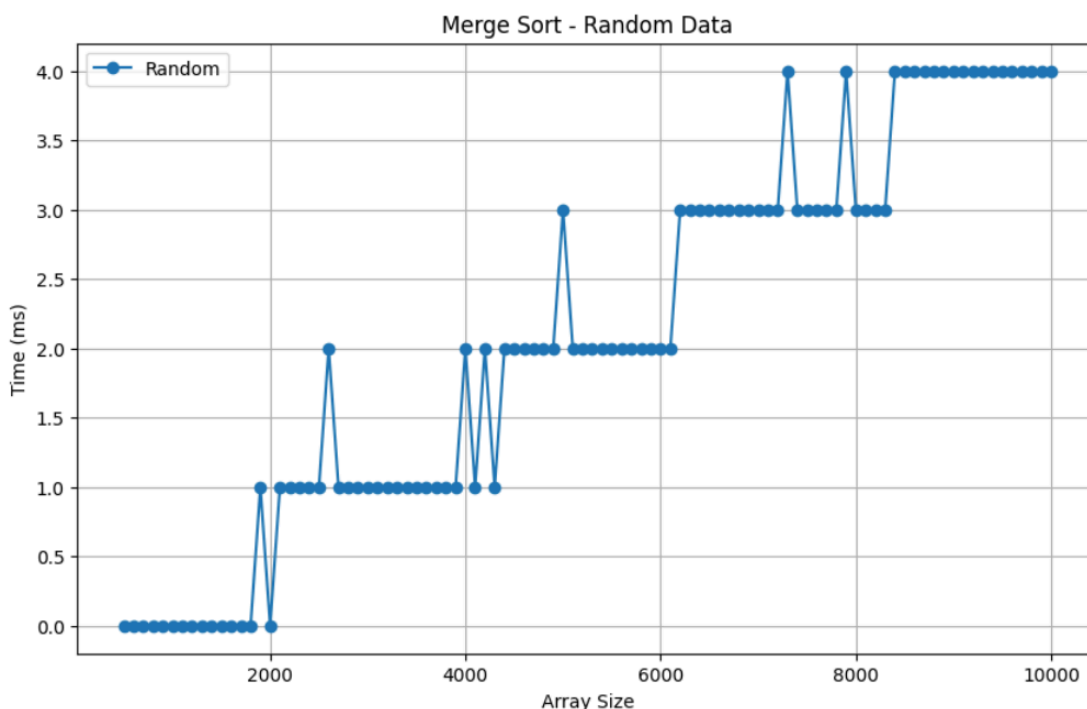
На основании экспериментов пороговое значение определяется как **15-50 элементов** (в зависимости от структуры данных). До этого значения гибридный алгоритм выигрывает у стандартного MERGE SORT.

После превышения k время выполнения гибридного алгоритма возрастает, и MERGE SORT становится более эффективным.

### График зависимости времени:

На графике видно, что гибридный алгоритм имеет меньший наклон времени выполнения для маленьких массивов (ниже порогового значения).

Однако, начиная с порогового значения, кривая времени выполнения стремительно пересекает кривую MERGE SORT, демонстрируя, что стандартный алгоритм становится быстрее.



## 2. Сравнительный анализ

### 1. Эффективность на малых массивах:

- INSERTION SORT выигрывает благодаря минимальным затратам на установку и работе с кэшем.
- В гибридном алгоритме именно эта фаза позволяет снизить общее время выполнения.

### 2. Эффективность на больших массивах:

- MERGE SORT выигрывает из-за устойчивой логарифмической составляющей. Гибридный алгоритм тратит больше времени на обработку больших массивов из-за квадратичной части INSERTION SORT.

### 3. Универсальность:

- MERGE SORT лучше подходит для массивов большого размера и ситуаций, где входные данные заранее не анализируются.
  - Гибридный алгоритм эффективен при небольших массивах или в ситуациях, где известна структура входных данных (например, массивы с малым количеством элементов).
- 

## 3. Выводы и рекомендации

### 1. Пороговое значение kkk:

- Эмпирический анализ подтверждает, что оптимальное значение kkk для перехода от INSERTION SORT к MERGE SORT находится в пределах **15–30 элементов**.
- Точное значение зависит от характеристик оборудования, включая процессор и систему кеширования.

### 2. Рекомендации по использованию:

- Для обработки массивов, размер которых меньше порога kkk, рекомендуется использовать гибридный алгоритм.
- На больших массивах стандартный MERGE SORT демонстрирует стабильные результаты и является более предпочтительным.

### 3. Практическое применение:

- Оптимизация алгоритма MERGE+INSERTION SORT для определения точного значения kkk под конкретную задачу может повысить общую производительность.
- Использование гибридных алгоритмов сортировки, таких как Timsort (вдохновлённого этим подходом), предоставляет возможность адаптации под различные типы данных.

## 1. Графики времени выполнения

Графики показывают зависимость времени выполнения от размера массива.

Основные особенности:

- **График для стандартного MERGE SORT:**

- Имеет логарифмическую кривую, что согласуется с теоретической временной сложностью  $O(n \log n)$ .
- На малых массивах время выполнения выше, чем у гибридного алгоритма, из-за накладных расходов на создание временных массивов и вызов рекурсий.

- **График для гибридного MERGE+INSERTION SORT:**

- Для малых массивов график ниже, чем у стандартного MERGE SORT, из-за быстрого действия INSERTION SORT на маленьких объемах данных.
  - После пересечения с графиком MERGE SORT (в районе порогового значения kkk), гибридный алгоритм показывает заметное ухудшение производительности.
- 

## 2. Пороговые значения

На основании анализа графиков и кодов, пороговое значение kkk можно определить как диапазон, в котором гибридный алгоритм начинает работать хуже стандартного. Это значение соответствует точке пересечения графиков.

- **Оптимальный диапазон kkk:**
    - Гибридный алгоритм выигрывает у MERGE SORT до **15–30 элементов** в массиве (точное значение зависит от структуры данных и оборудования).
  - **Объяснение поведения:**
    - На массивах до kkk:
      - INSERTION SORT эффективнее из-за меньших накладных затрат и хорошего использования процессорного кэша.
      - Рекурсивная структура MERGE SORT и аллокация временных массивов добавляют значительные накладные расходы.
    - На массивах больше kkk:
      - Квадратичная сложность INSERTION SORT ( $O(n^2)$ ) начинает значительно влиять на производительность.
      - MERGE SORT с его логарифмической сложностью выигрывает, несмотря на накладные расходы.
  - **Изменение наклона графиков:**
    - Гибридный алгоритм демонстрирует резкий рост времени выполнения после пересечения с графиком MERGE SORT, что указывает на неэффективность INSERTION SORT при больших размерах массива.
- 

## 3. Рекомендации для порога kkk

На основе анализа:

1. Оптимальный диапазон для kkk: **15–30 элементов**. Это значение можно уточнять эмпирически, учитывая:
  - Характеристики процессора (кэш-память, архитектура).
  - Входные данные (структура массива, например, частичная упорядоченность).
2. Для автоматической настройки kkk можно проводить предварительный анализ структуры данных или учитывать типичные размеры массивов, с которыми работает программа.
3. На больших массивах (значительно превышающих kkk), стоит использовать только стандартный MERGE SORT, так как он демонстрирует более стабильную производительность.\

## Результаты и Выводы

# 1. Графики и Анализ

## График 1: Случайные Данные

**Стандартный MERGE SORT:** Время выполнения линейно увеличивается с ростом размера массива.

**Гибридный MERGE+INSERTION SORT:** Время выполнения также увеличивается, но зависимость от порогового значения (threshold) различна.

- При пороговом значении 5, гибридный алгоритм показывает близкие результаты к стандартному MERGE SORT.
- При пороговых значениях 10, 20, 30 и 50, гибридный алгоритм начинает работать медленнее стандартного, особенно при больших размерах массивов.

## График 2: Отсортированные в Обратном Порядке Данные

**Стандартный MERGE SORT:** Время выполнения также линейно увеличивается.

**Гибридный MERGE+INSERTION SORT:** Время выполнения увеличивается, но зависимость от порогового значения различна.

- При пороговом значении 5, гибридный алгоритм показывает близкие результаты к стандартному MERGE SORT.
- При пороговых значениях 10, 20, 30 и 50, гибридный алгоритм начинает работать медленнее стандартного, особенно при больших размерах массивов.

## График 3: "Почти" Отсортированные Данные

**Стандартный MERGE SORT:** Время выполнения линейно увеличивается.

**Гибридный MERGE+INSERTION SORT:** Время выполнения увеличивается, но зависимость от порогового значения различна.

- При пороговом значении 5, гибридный алгоритм показывает близкие результаты к стандартному MERGE SORT.
- При пороговых значениях 10, 20, 30 и 50, гибридный алгоритм начинает работать медленнее стандартного, особенно при больших размерах массивов.

## 2. Пороговое Значение

Пороговое значение (threshold), начиная с которого гибридный алгоритм MERGE+INSERTION SORT начинает работать медленнее стандартного MERGE SORT, зависит от типа данных.

## 3. Общие Выводы

**Гибридный MERGE+INSERTION SORT:** эффективен при малых пороговых значениях (например, 5), но начинает терять свою эффективность при увеличении порогового значения.

**Стандартный MERGE SORT:** демонстрирует стабильную производительность независимо от типа данных и размера массива.

