



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

Факультет компьютерных наук  
Департамент программной инженерии

# КОМПОНЕНТ-РАСШИРЕНИЕ РСУБД SQLITE ДЛЯ ИНДЕКСИРОВАНИЯ ДАННЫХ МОДИФИКАЦИЯМИ В- ДЕРЕВЬЕВ

Антон Ригин

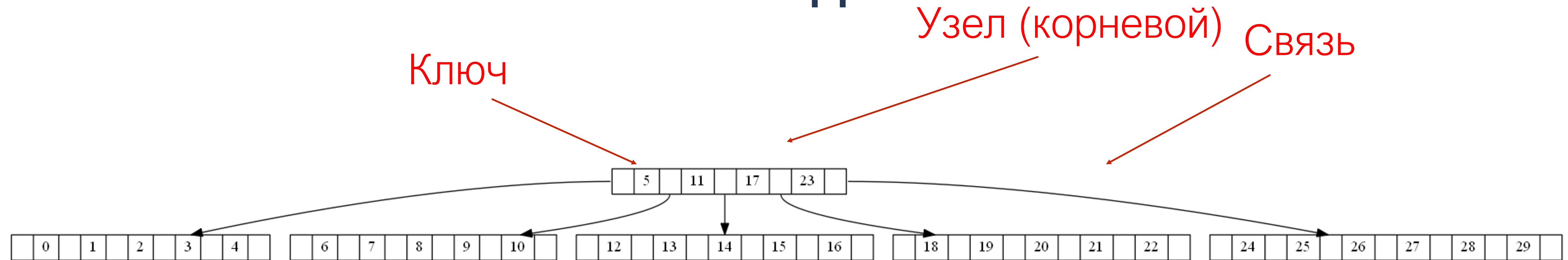
Научный руководитель: Шершаков С.А.,  
ст. преп. ДПИ ФКН, н.с. НУЛ ПОИС ФКН

Москва, 2019

# ПРЕДМЕТНАЯ ОБЛАСТЬ

- Сильно ветвящиеся деревья: В-деревья,  $V^+$ -деревья,  $V^*$ -деревья и  $V^{*+}$ -деревья
- Эмпирическая оценка сложности основных операций и объёма занимаемой памяти
- Индексы в СУБД
- SQLite – встраиваемая реляционная СУБД
- Использование сильно ветвящихся деревьев в качестве индекса в SQLite

# ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ



*Пример B-дерева порядка 6, высоты 2*

Узел (листовой)

# ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

- **Сильно ветвящееся дерево** (структура данных) – такое дерево, которое содержит в одном узле более одного элемента с ключом и более одной связи с дочерними узлами.
- **В-дерево** – Сильно ветвящееся дерево. В-дерево построено так, что если некоторый узел содержит  $k$  ключей, то у данного узла  $k + 1$  потомков, и для любого  $i$ , такого, что  $1 \leq i \leq k + 1$ , верно, что все ключи в  $i$ -м потомке данного узла не меньше, чем  $i$ -й ключ данного узла, и не больше, чем  $i + 1$ -й ключ данного узла [6].
- **Порядок В-дерева** – такое число  $t$ , что для любого некорневого узла дерева верно неравенство:  $t - 1 \leq k \leq 2t - 1$ , где  $k$  – число ключей в узле. Корневой узел для непустого В-дерева содержит  $1 \leq k \leq 2t - 1$  ключей, для пустого В-дерева – 0 ключей [6].
- В-дерево является **сбалансированным деревом**, поэтому его высота будет равна  $O(\log_t n)$ , где  $n$  – число ключей в дереве [6].

[6] Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. Алгоритмы: построение и анализ. 3-е изд.

# ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

- **B<sup>+</sup>-дерево** – модификация B-дерева. В B<sup>+</sup>-дереве настоящие ключи хранятся лишь в листьях дерева, а во внутренних узлах хранятся лишь ключи-маршрутизаторы, необходимые для поиска по дереву. Листья в B<sup>+</sup>-дереве содержат  $t \leq n \leq 2t$  ключей, где  $t$  – порядок дерева, ограничения для внутренних узлов такие же, как и в B-дерева. Деление листьев происходит поровну на две части, крайний ключ из левой половины делимого узла копируется в родительскую вершину в качестве ключа-маршрутизатора аналогично перемещению медианы для обычного деления, деление внутренних узлов происходит так же, как и в B-дерева [3].
- **B<sup>\*</sup>-дерево** – модификация B-дерева. Каждый узел заполняется не менее, чем на 2/3, а не 1/2. По этой причине, вместо традиционного разбиения узла, происходит

перераспределение ключей между соседними узлами-потомками, либо, если нет незаполненных соседей, то узел разбивается на три (а не на две) части [1] [5].

- **B<sup>+</sup>-дерево** – модификация B-дерева, **разработанная в рамках выполнения курсовой работы за 3 курс.** Представляет собой совмещение B<sup>+</sup>-дерева и B<sup>\*</sup>-дерева: модификация заполненных вершин при вставке выполняется как в B<sup>\*</sup>-дерева (переливание либо разбиение на три части), при этом, как в B<sup>+</sup>-дерева, реальные данные хранятся только в листьях, в остальных вершинах находятся лишь ключи-маршрутизаторы.

[1] <https://xlinux.nist.gov/dads/HTML/bstartree.html>

[3] Kerttu Pollari-Malmi. B<sup>+</sup>-trees:

<https://www.cs.helsinki.fi/u/mluukkai/tirak2010/B-tree.pdf>

[5] Д. Кнут. Искусство программирования. Том 3. 2-е изд.

# ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

- **СУБД** – система управления базами данных
- **РСУБД** – реляционная СУБД
- **SQLite** – РСУБД с открытым исходным кодом (написана на языке C)
- **Расширение SQLite** – библиотека динамического подключения, расширяющая функционал РСУБД SQLite и предоставляющая новые функции [4]



# АКТУАЛЬНОСТЬ

- В настоящее время растут объёмы обрабатываемых данных
- Необходимо разрабатывать новые эффективные подходы к индексации данных в СУБД
- СУБД SQLite содержит небольшое число способов индексирования данных – актуально добавление новых

# СУЩЕСТВУЮЩИЕ РЕШЕНИЯ

- В-дерево является способом индексации по умолчанию в SQLite
- Существует ряд расширений для SQLite, добавляющих, например, индексирование при помощи R-дерева
- Расширений с В+-деревом, В\*-деревом и В\*+-деревом не обнаружено

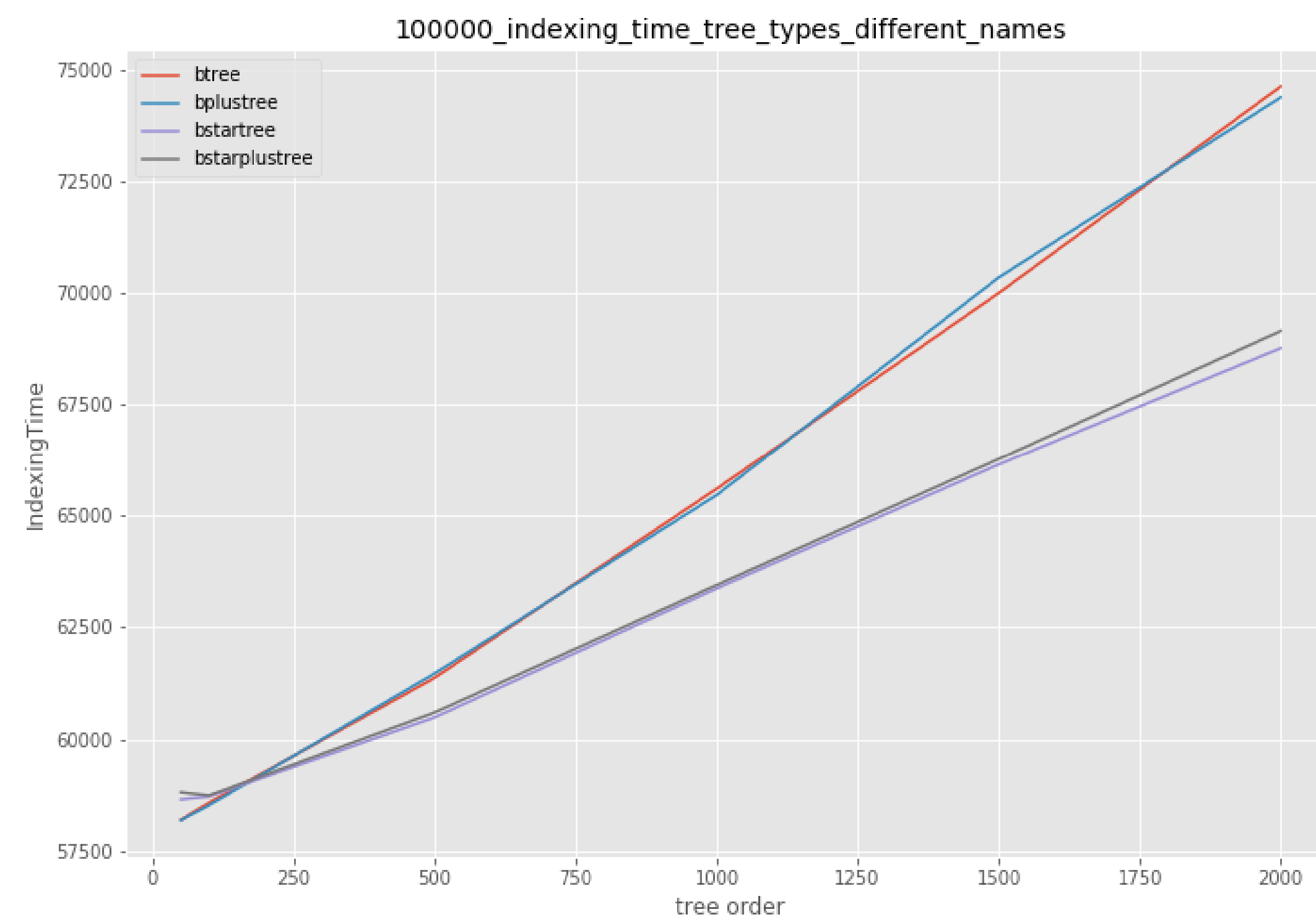
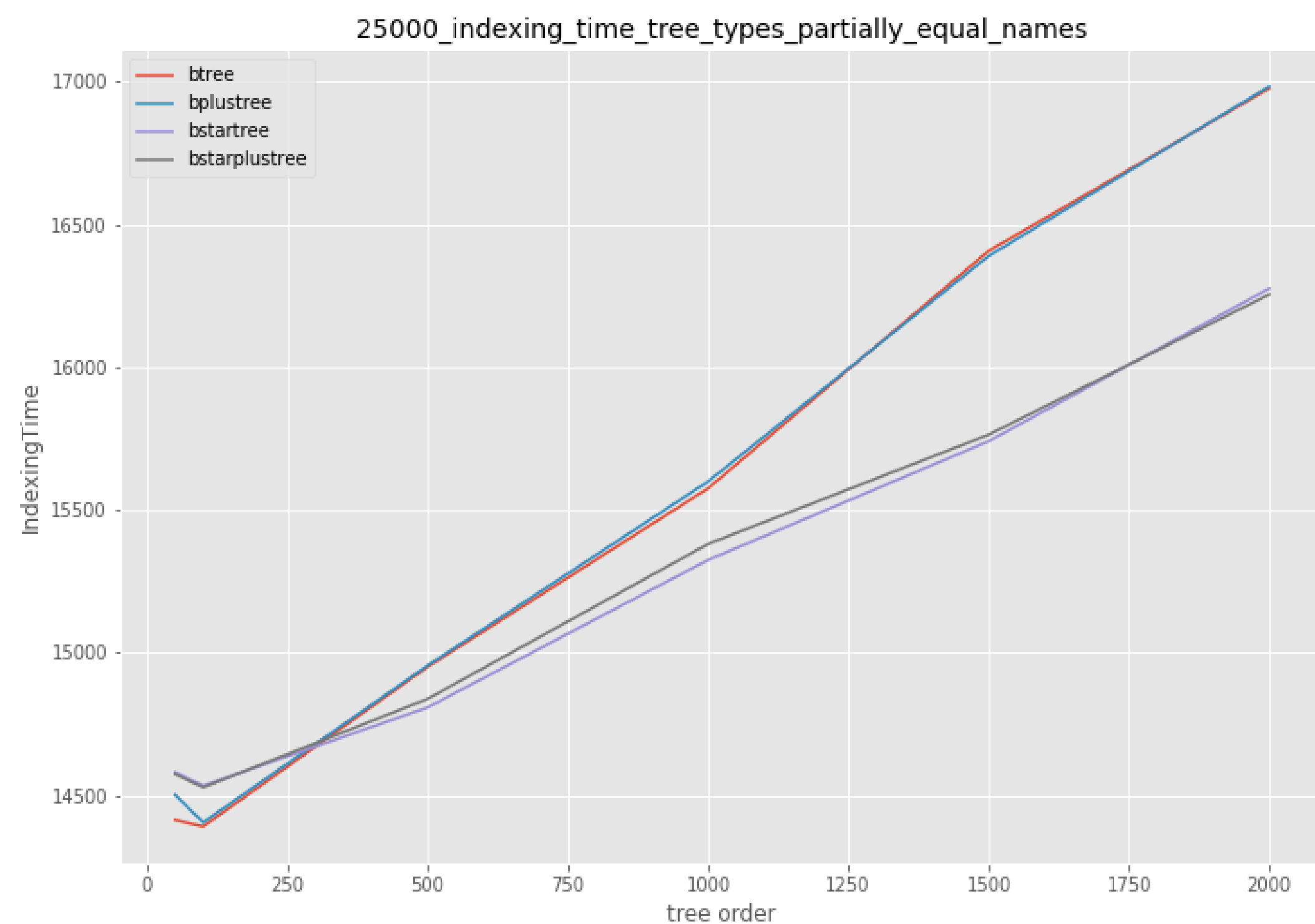


# ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

- **Цель работы**
  - ✓ Разработать расширение для SQLite, позволяющее использовать в качестве индекса в данной СУБД модификации В-дерева:  $B^+$ -дерево,  $B^*$ -дерево и  $B^{*+}$ -дерево
- **Задачи работы**
  - ✓ Подключить имеющуюся C++-библиотеку к SQLite в качестве расширения
  - ✓ Разработать алгоритм выбора наиболее подходящей структуры данных (В-дерева либо одной из его модификаций) для индексирования таблицы
  - ✓ Написать расширение для SQLite-менеджера для визуализации В-деревьев и их модификаций, а также рассчитанных для них метрик

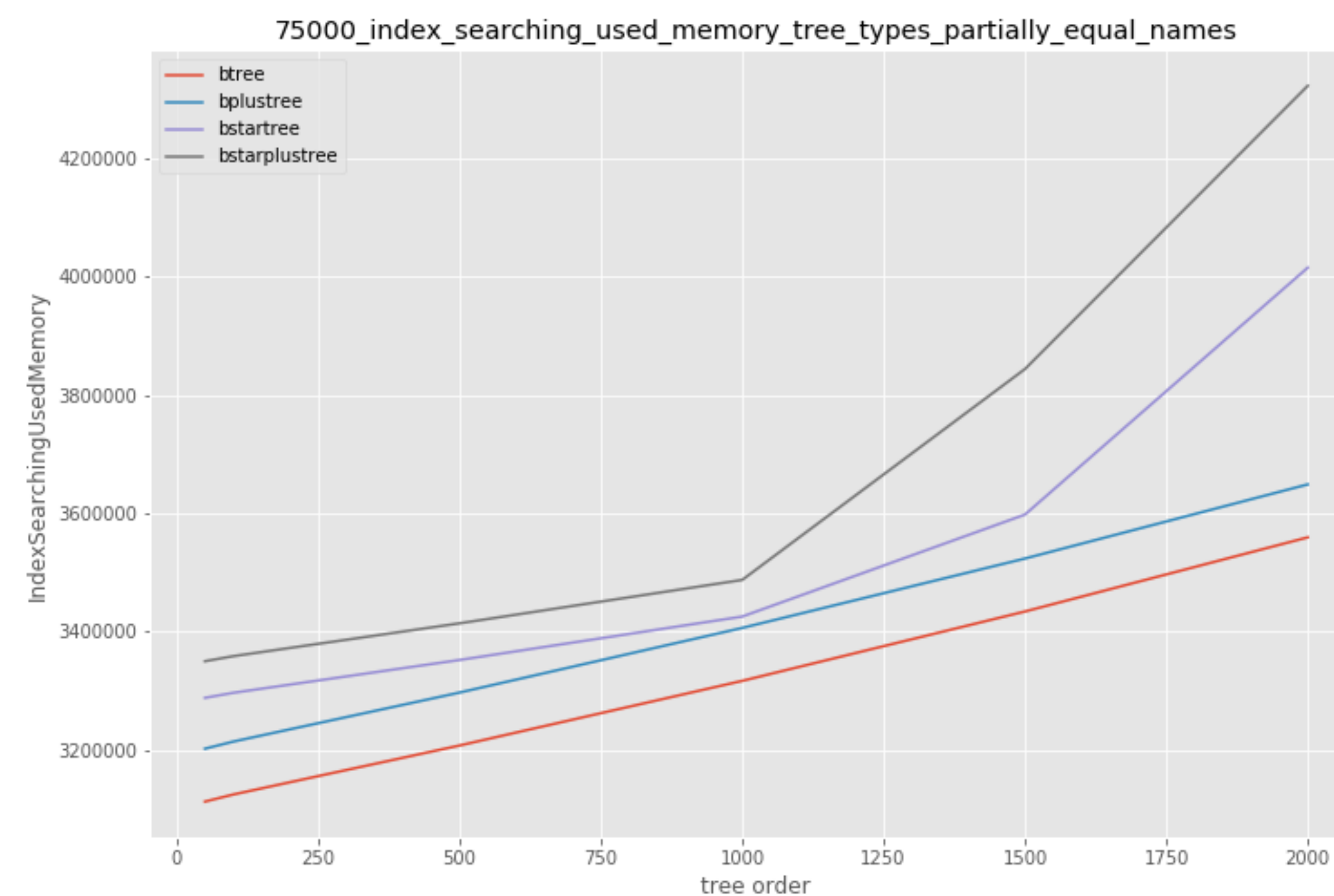
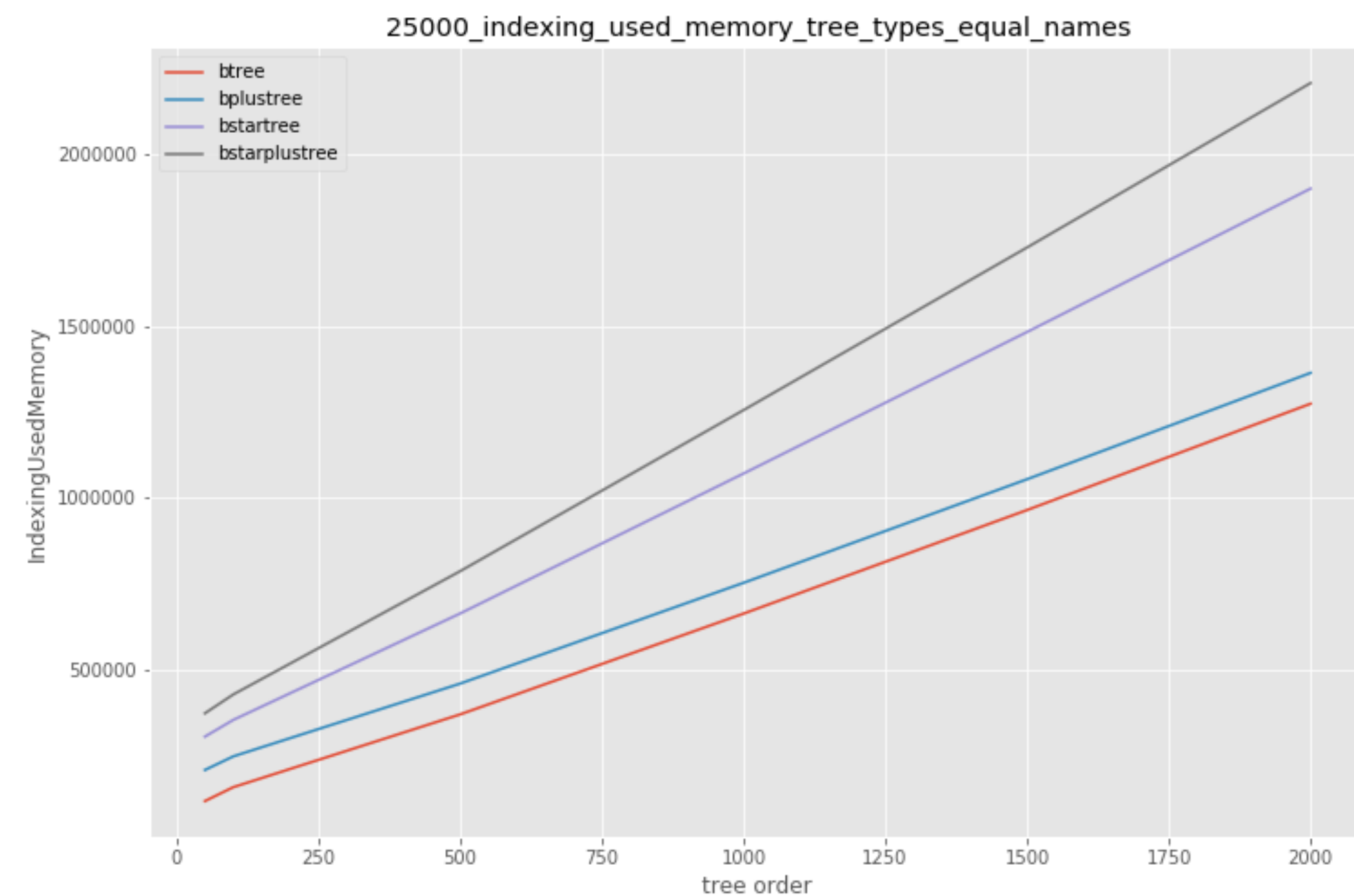
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

## Индексация – время



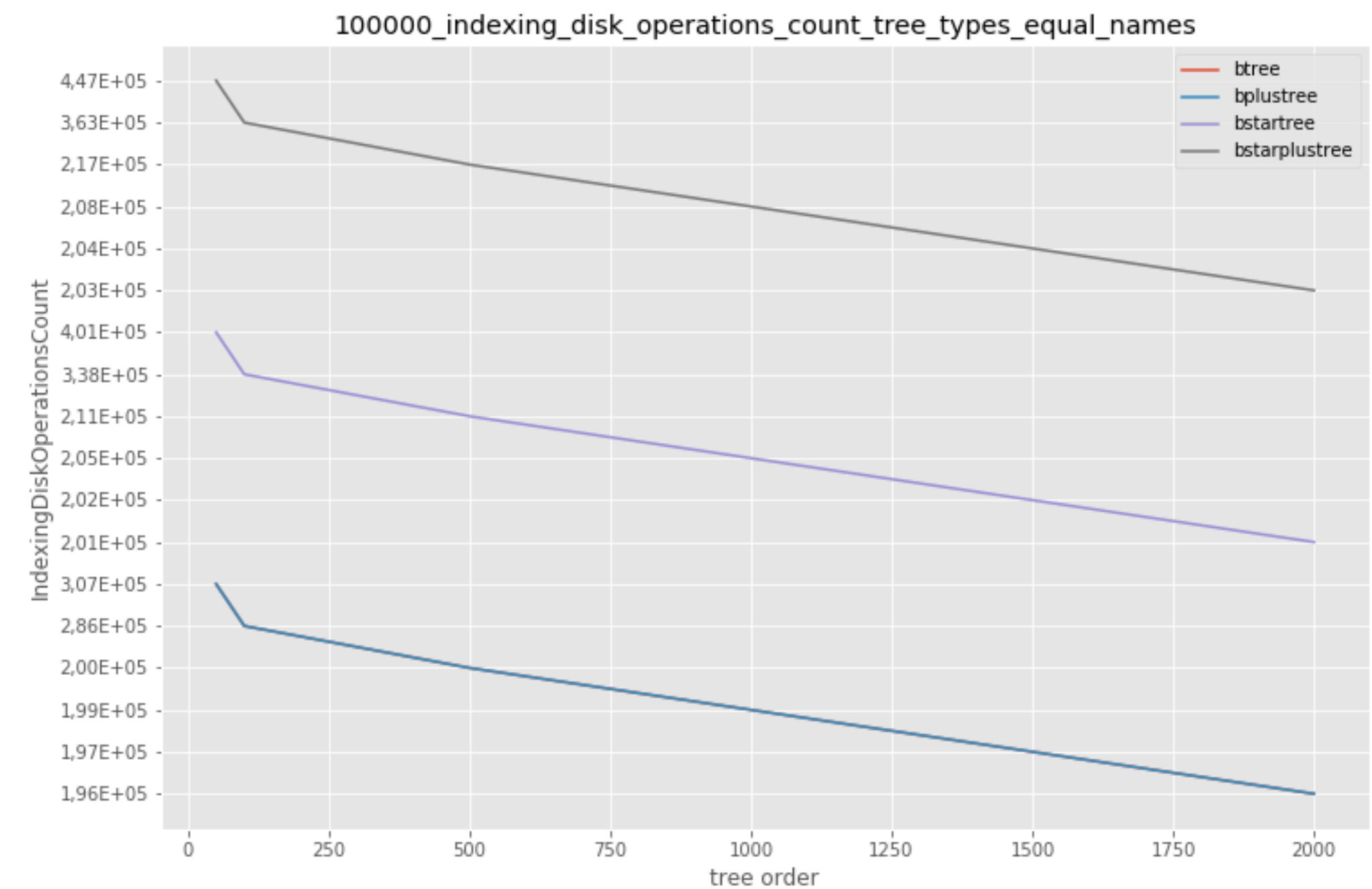
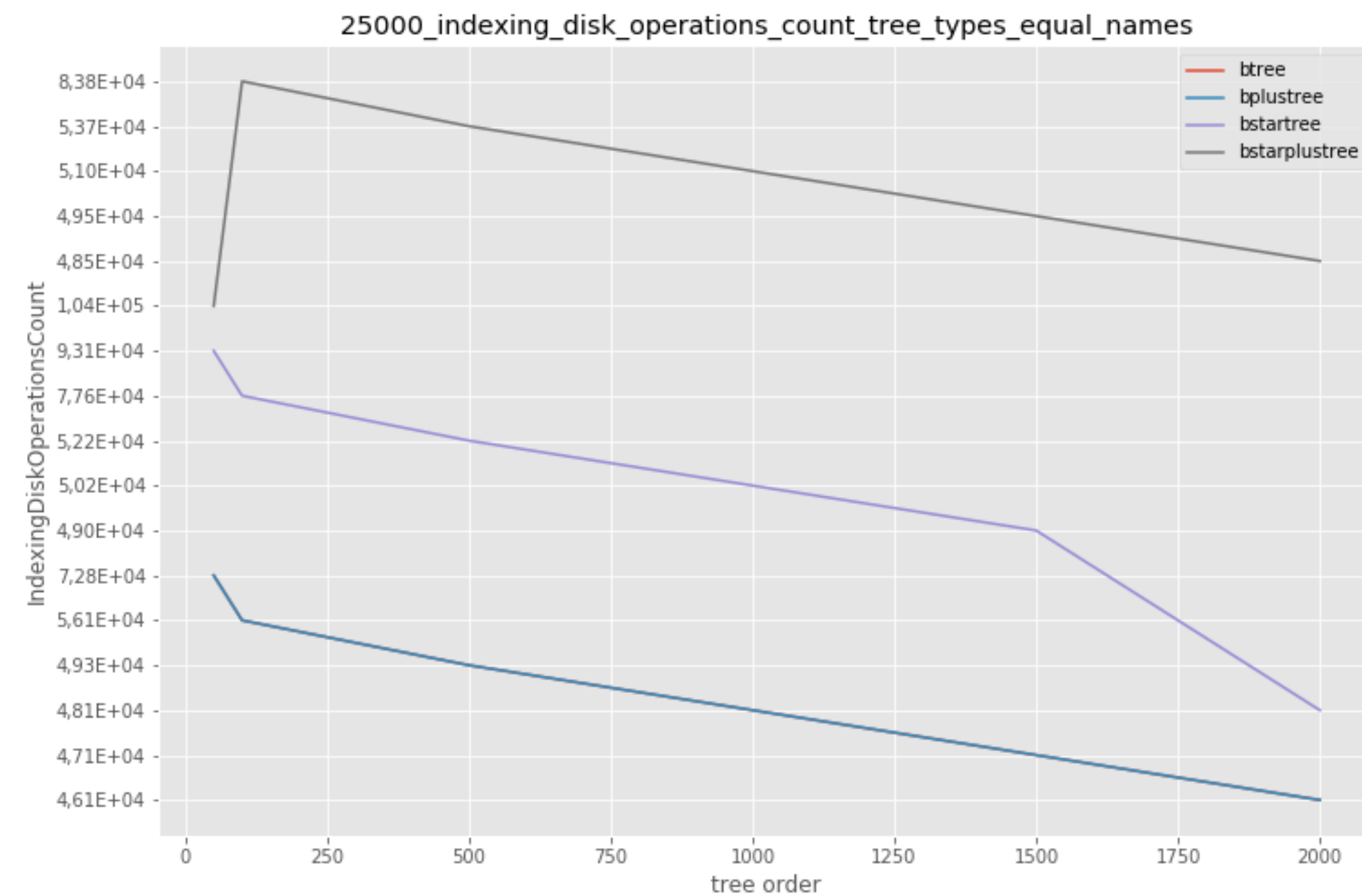
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

## Индексация – память



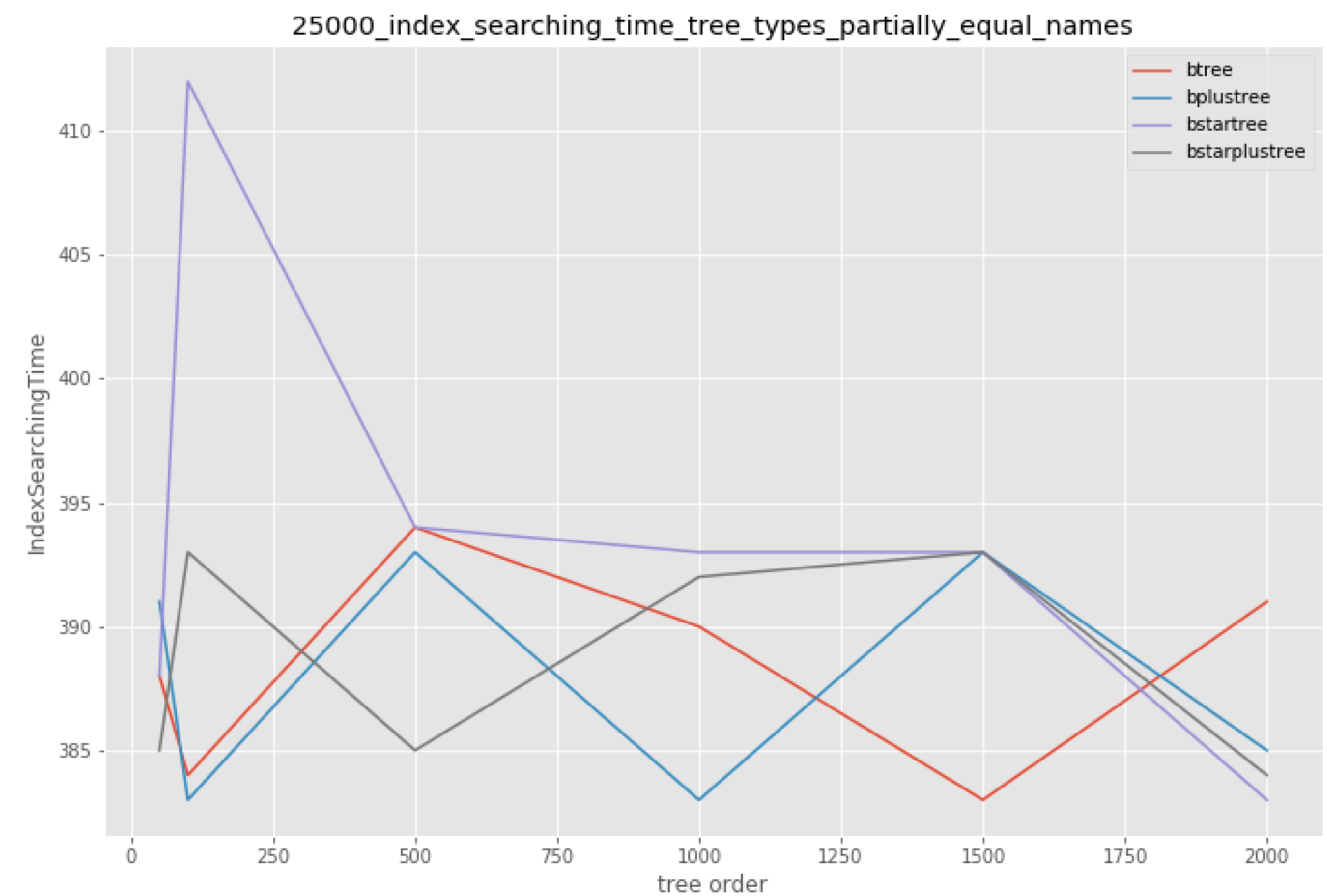
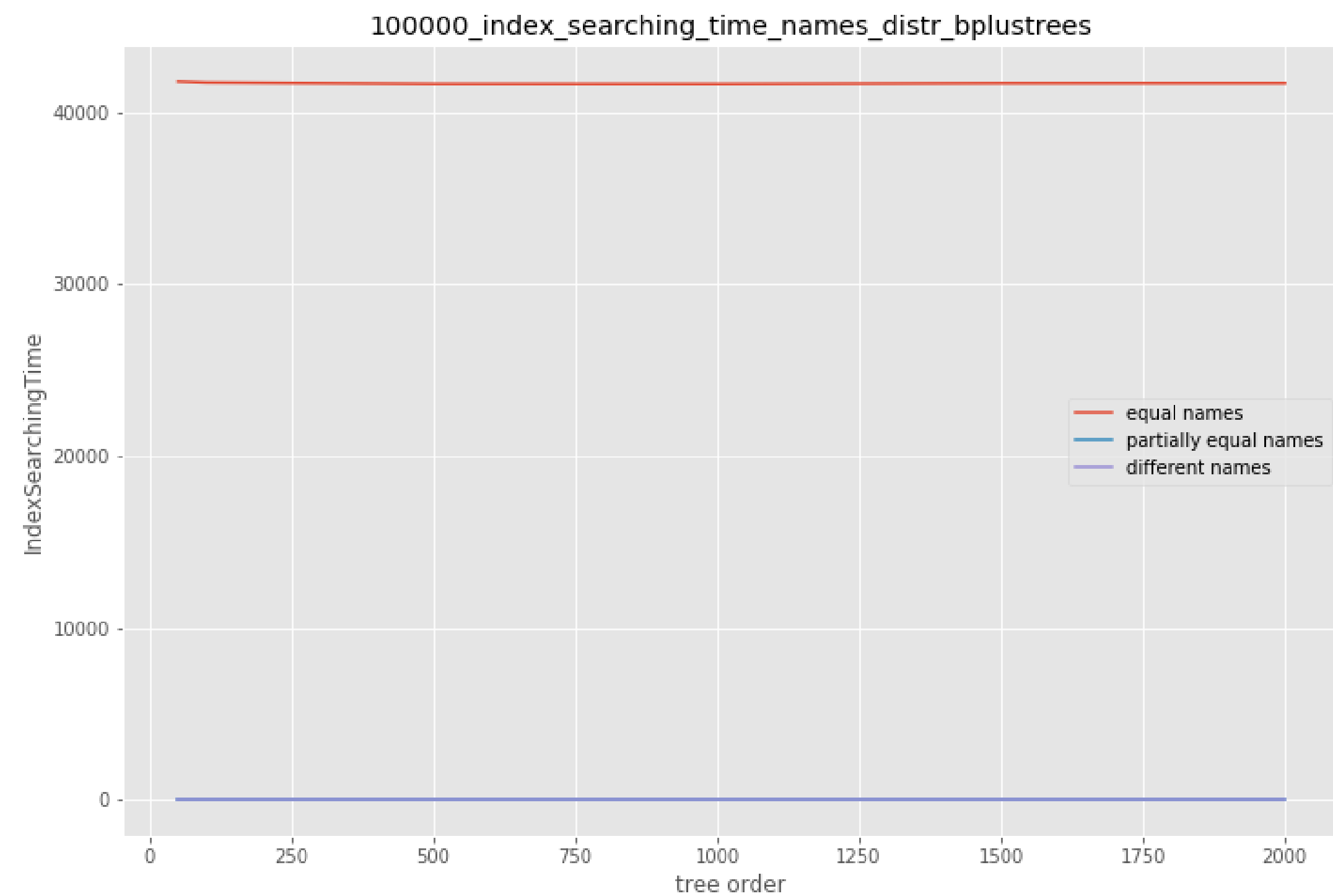
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

## Индексация – дисковые операции



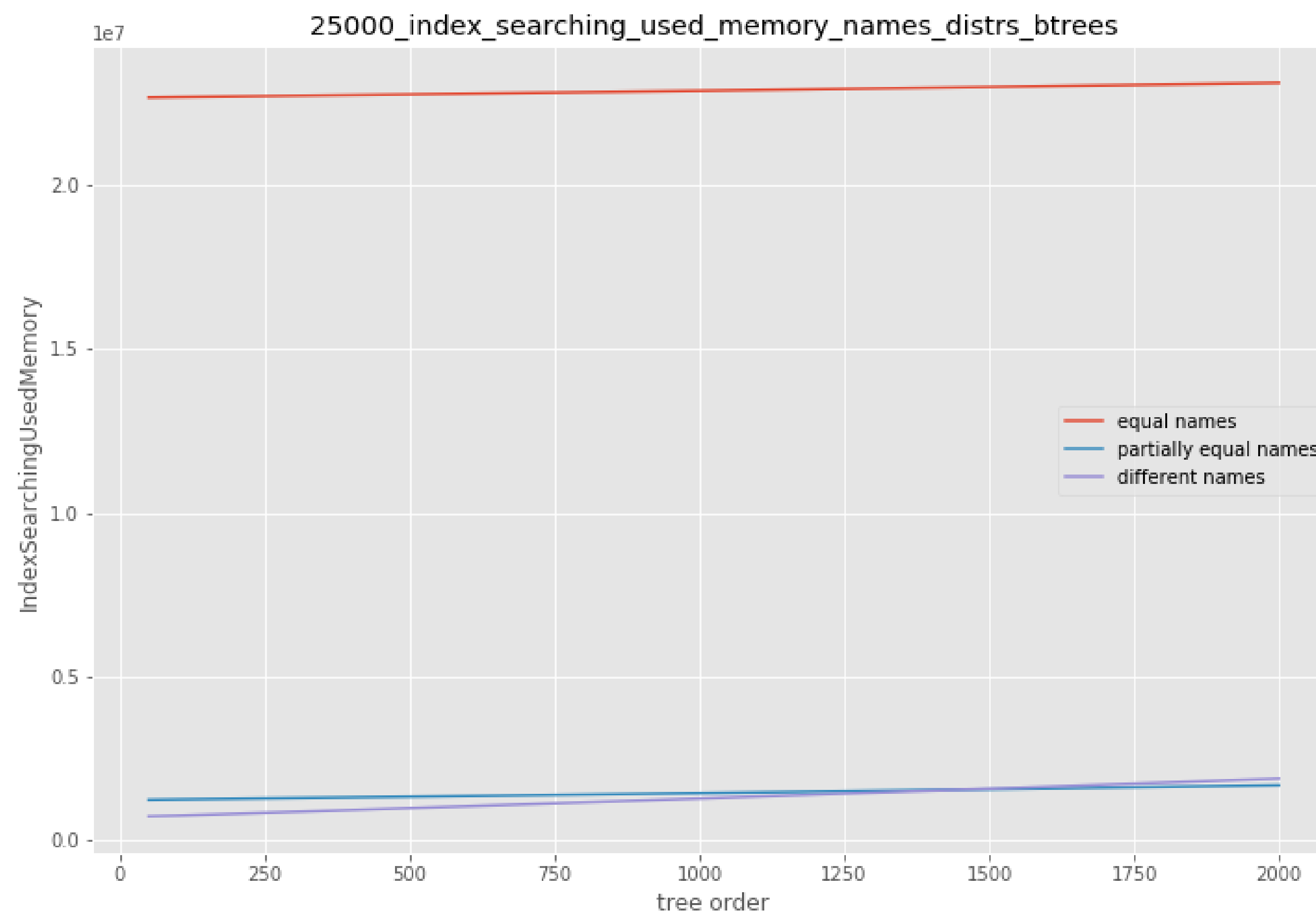
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Поиск по индексу – время



# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

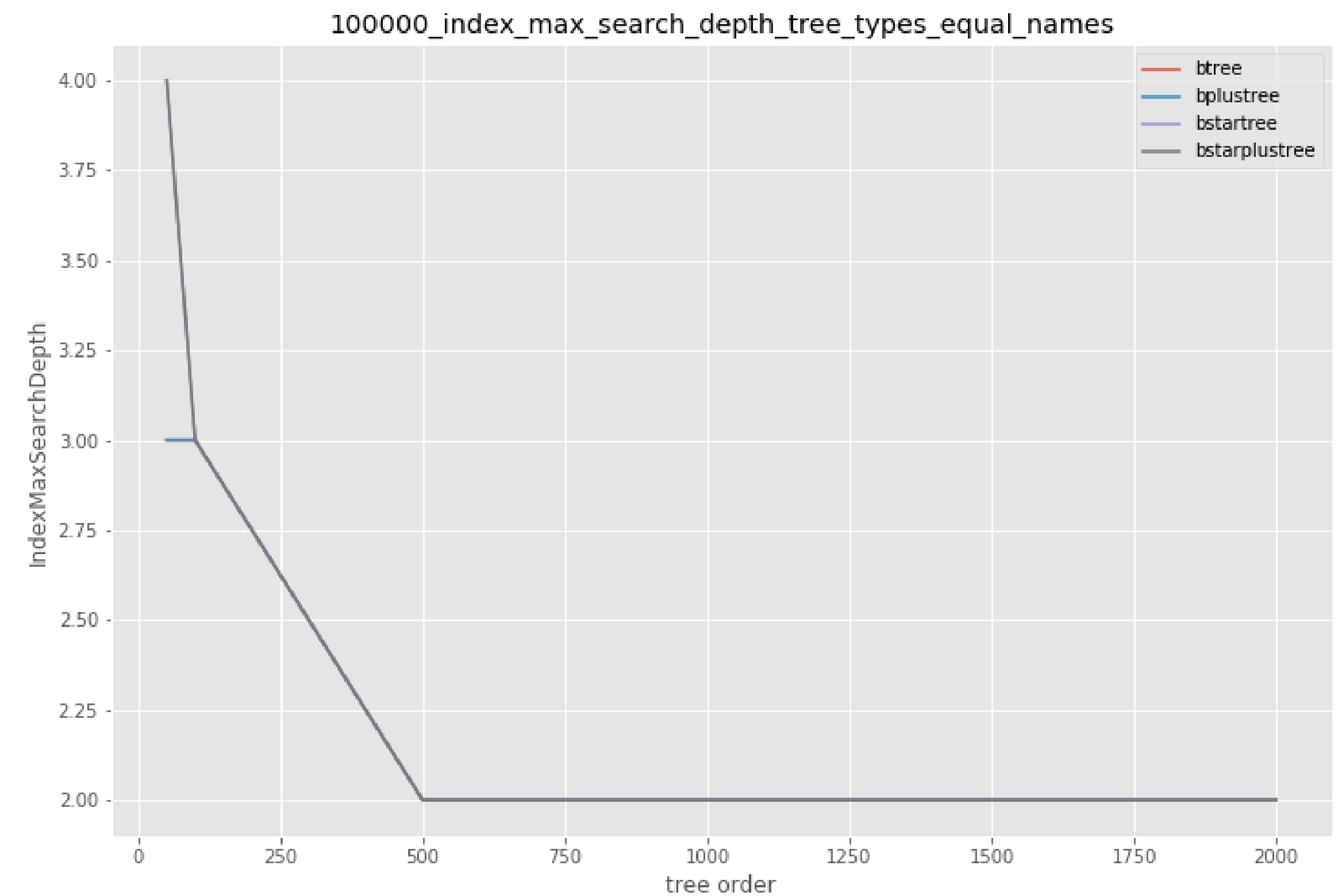
Поиск по индексу – память





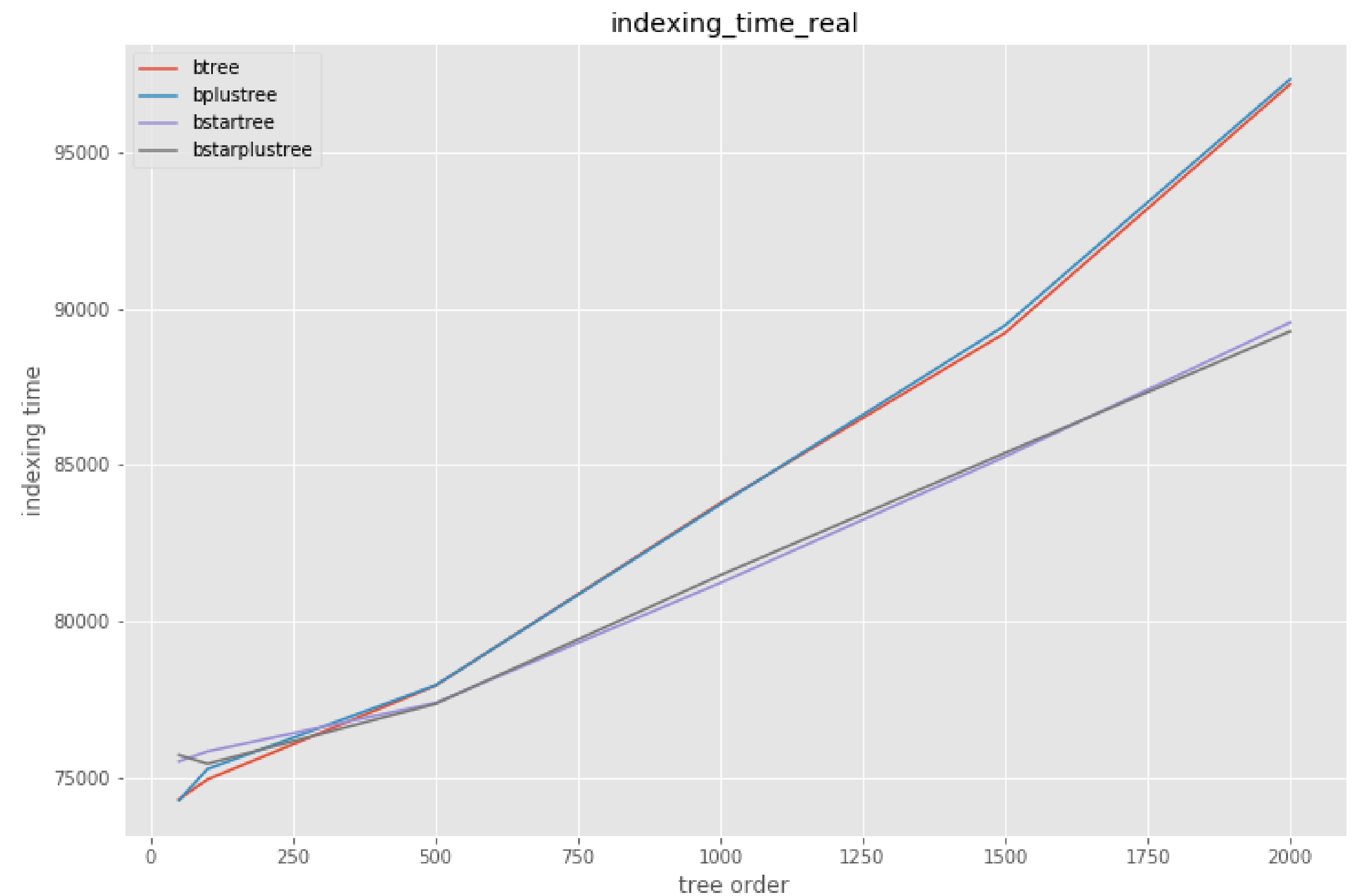
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Поиск по индексу – определение высоты дерева



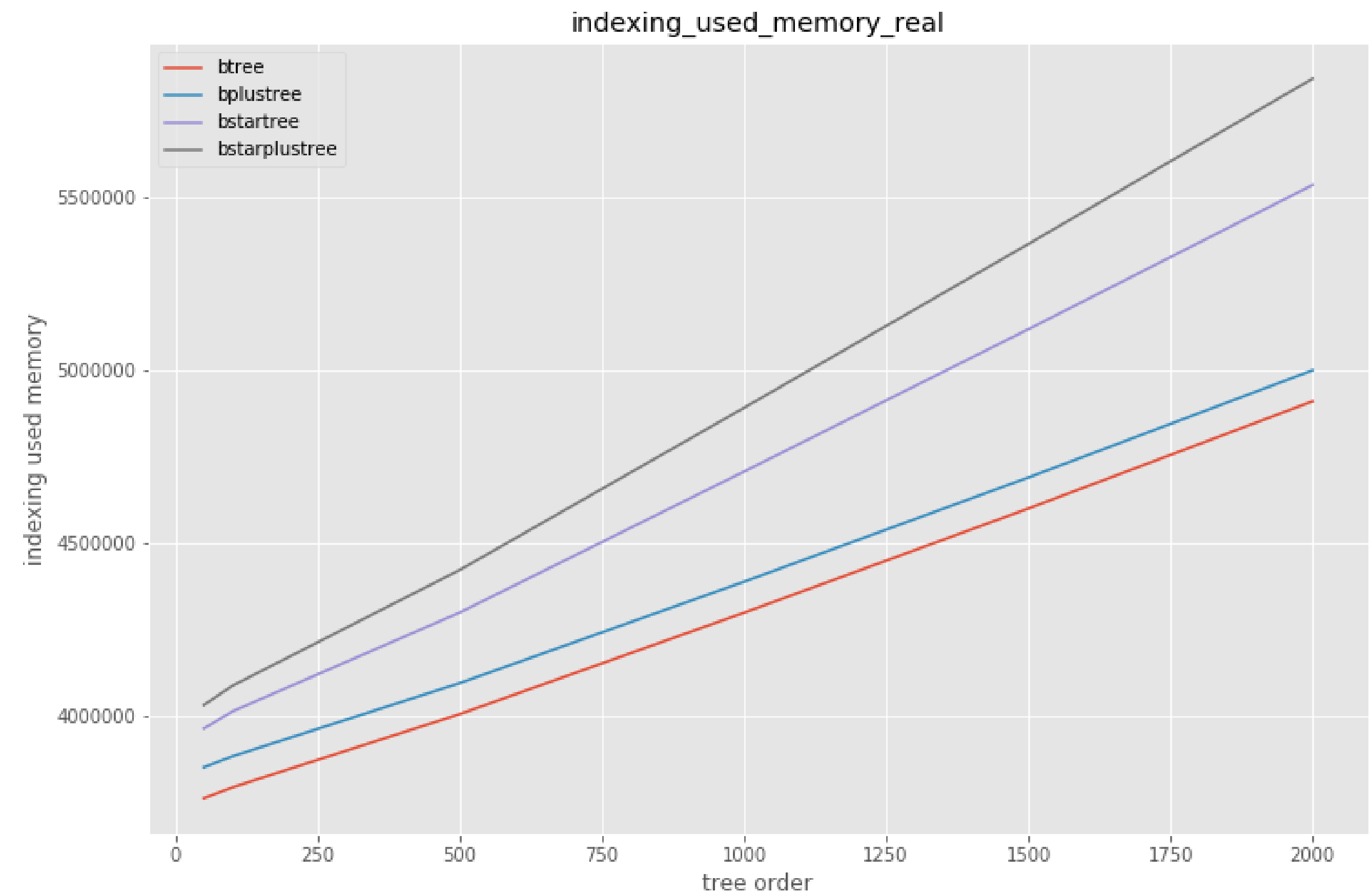
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Индексация (реальные данные) – время



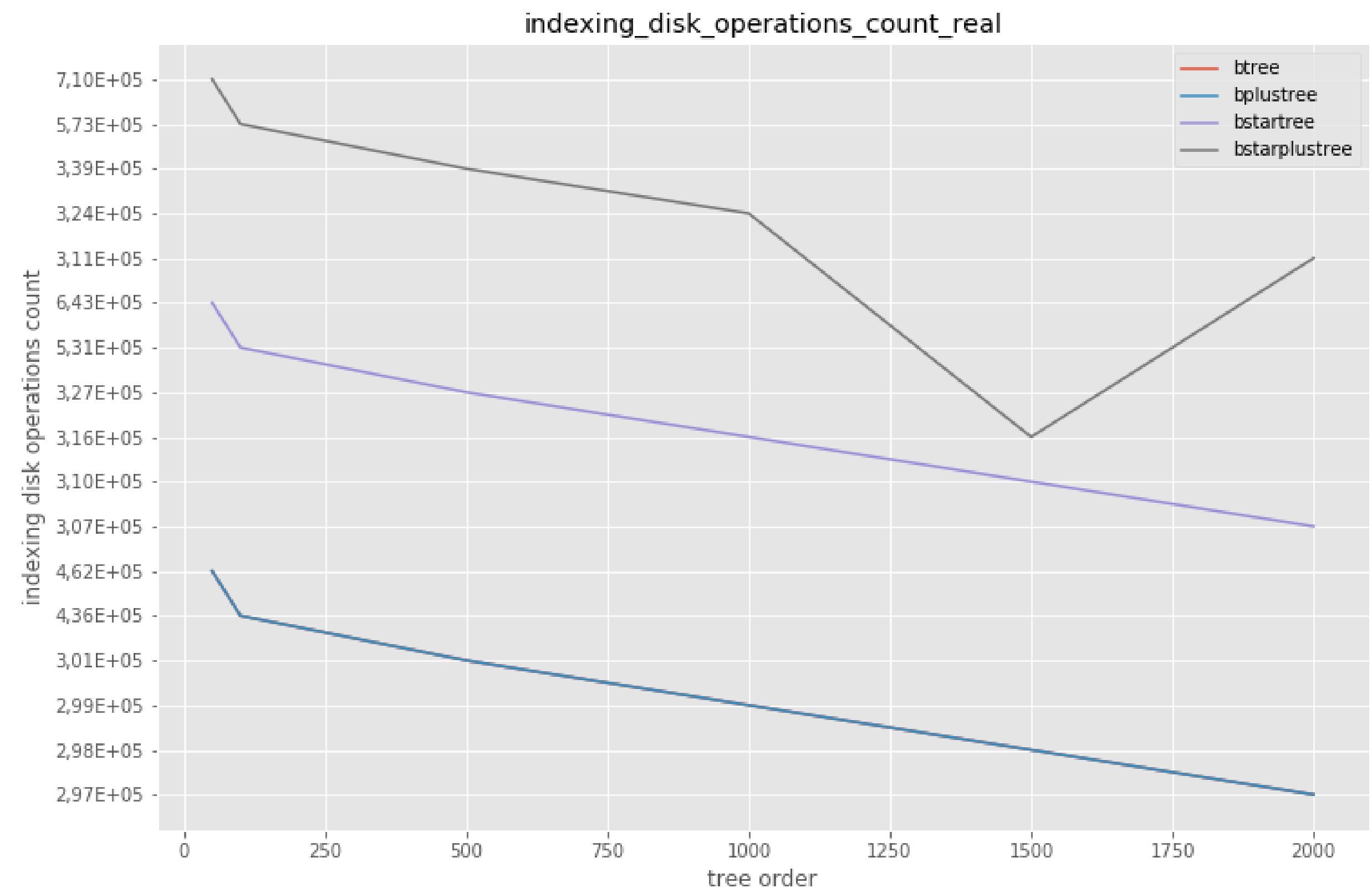
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Индексация (реальные данные) – память



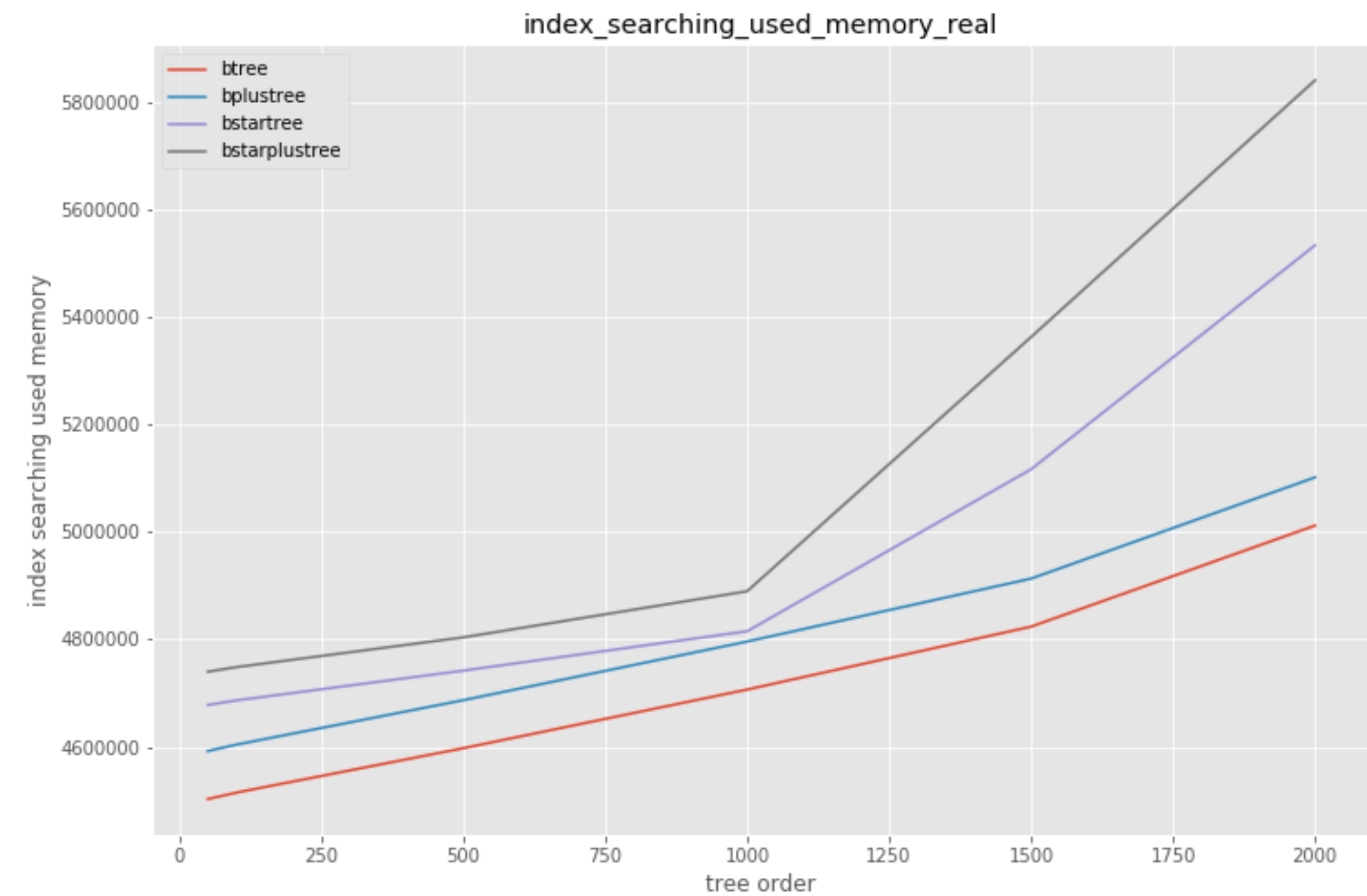
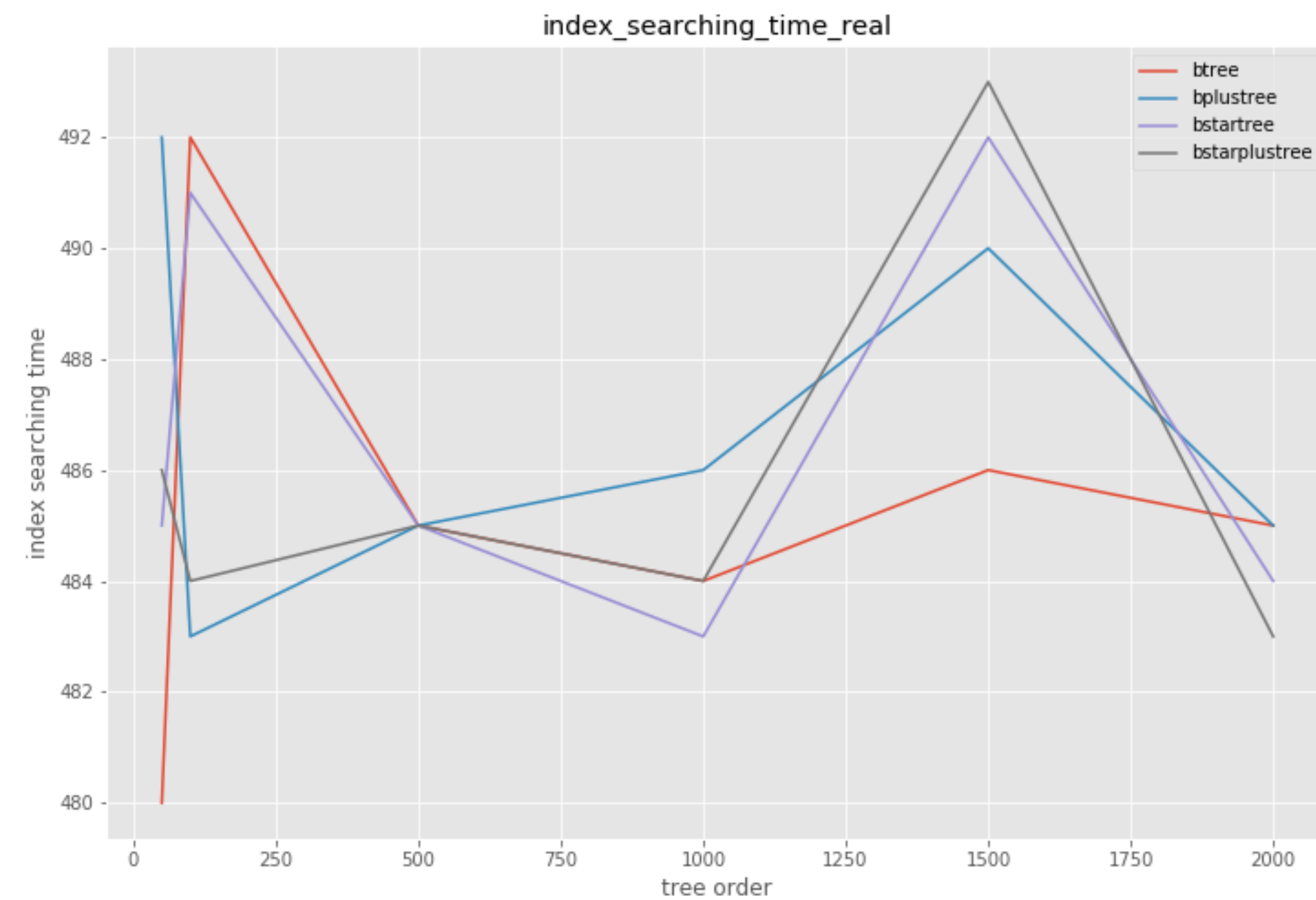
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Индексация (реальные данные) – дисковые операции



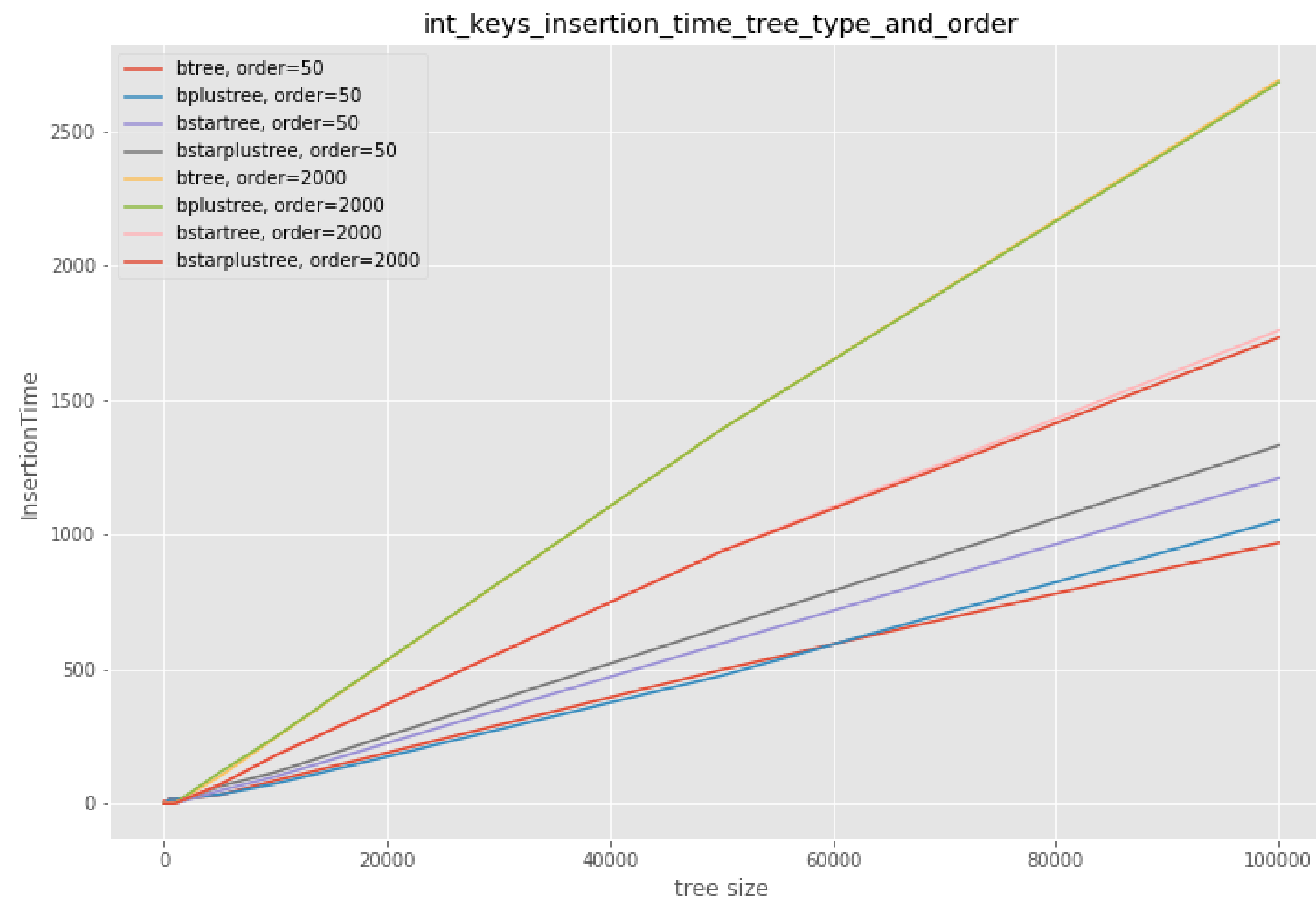
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Поиск по индексу (реальные данные) – время и память



# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Вставка в дерево (целочисленные ключи) – время

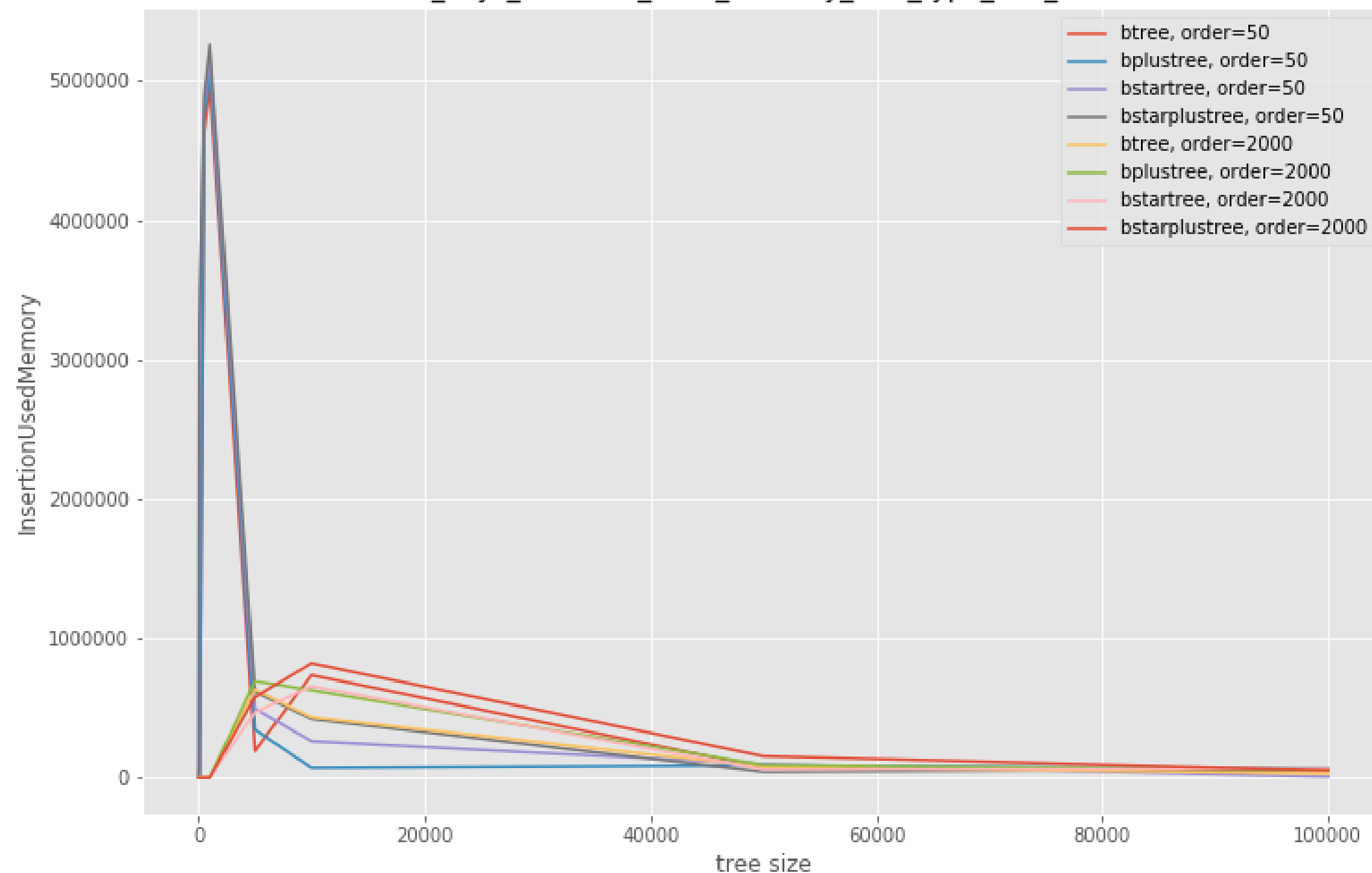




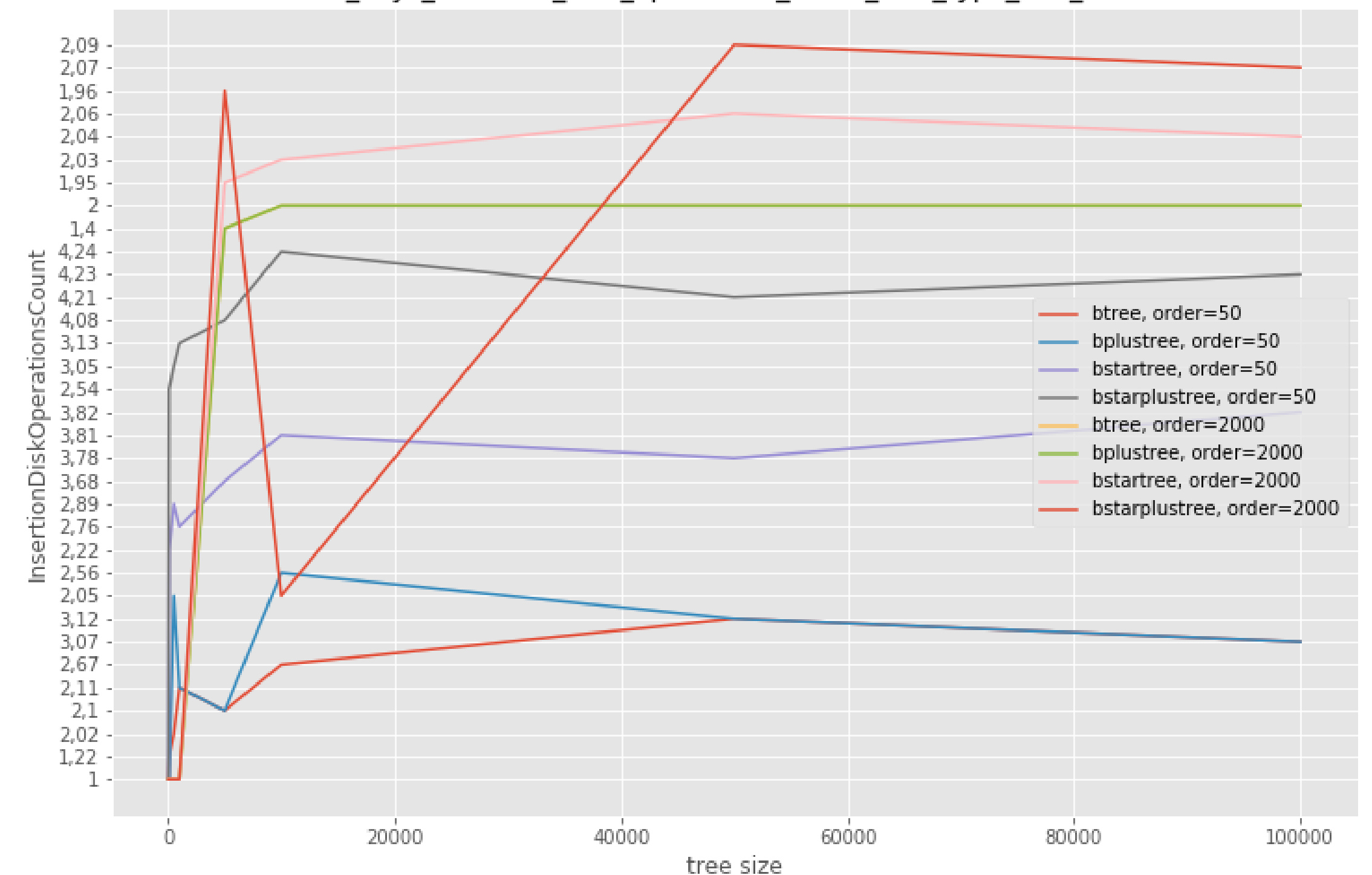
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Вставка в дерево (целочисленные ключи) – память и дисковые операции

int\_keys\_insertion\_used\_memory\_tree\_type\_and\_order

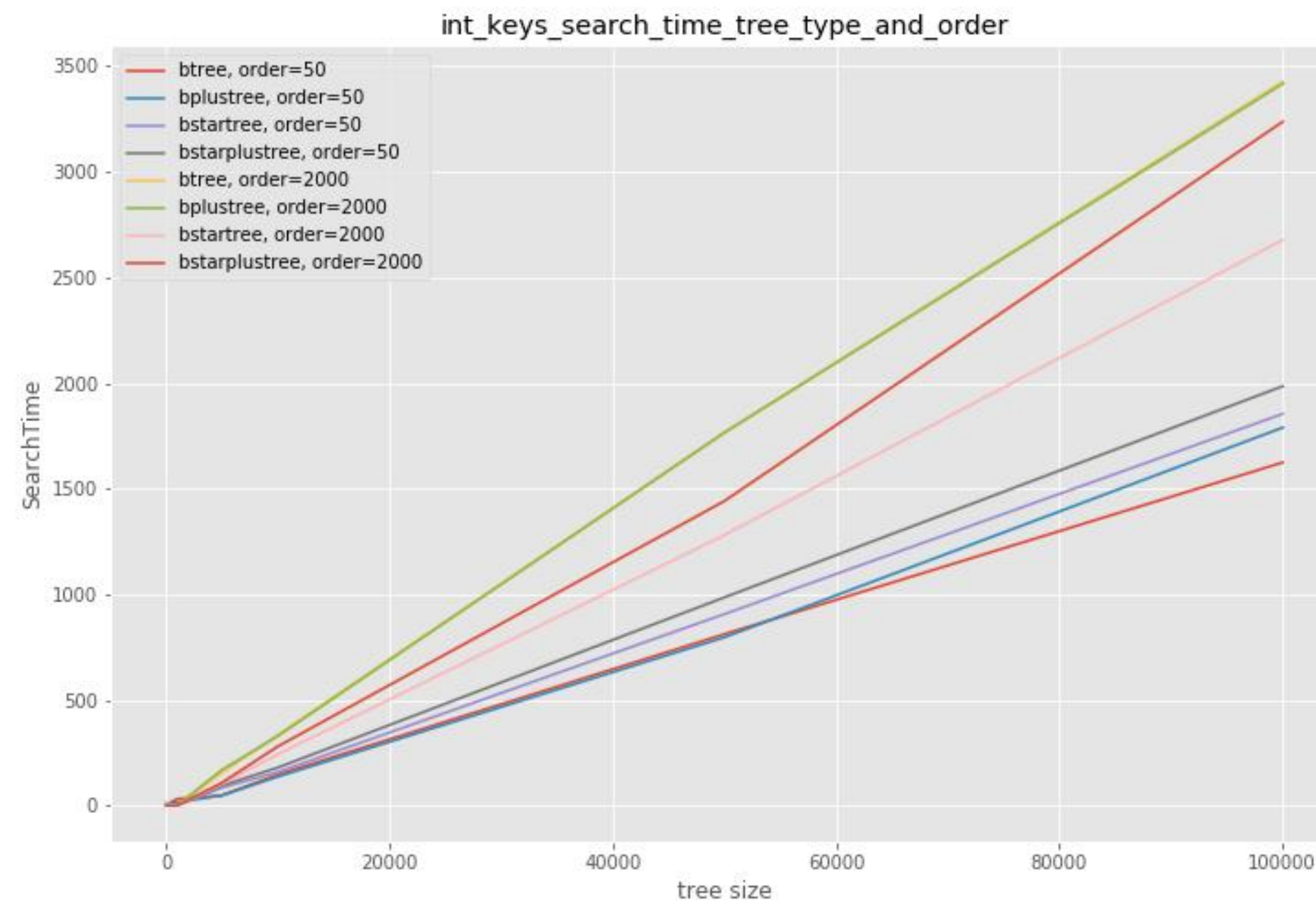


int\_keys\_insertion\_disk\_operations\_count\_tree\_type\_and\_order



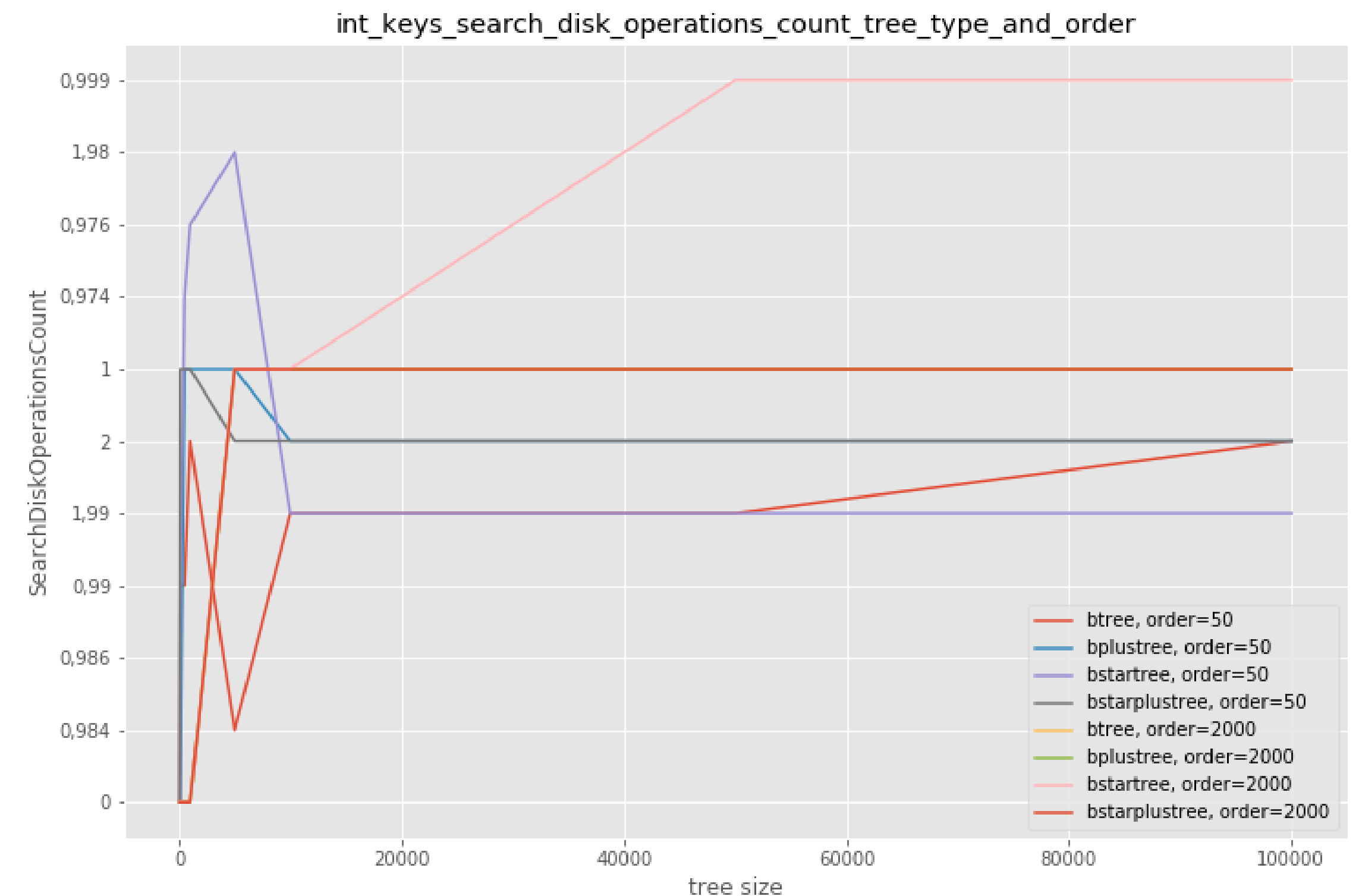
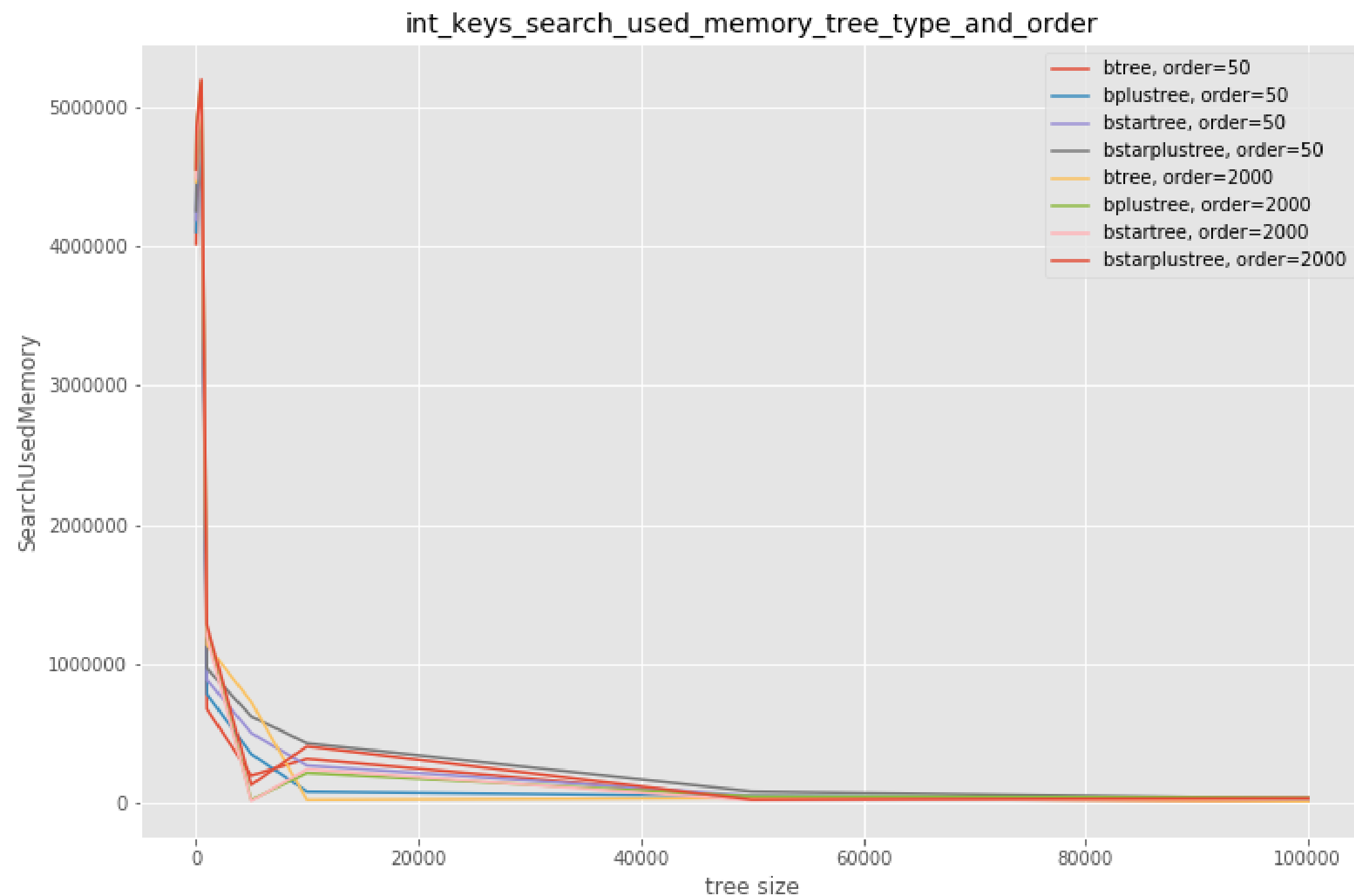
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Поиск в дереве (целочисленные ключи) – время



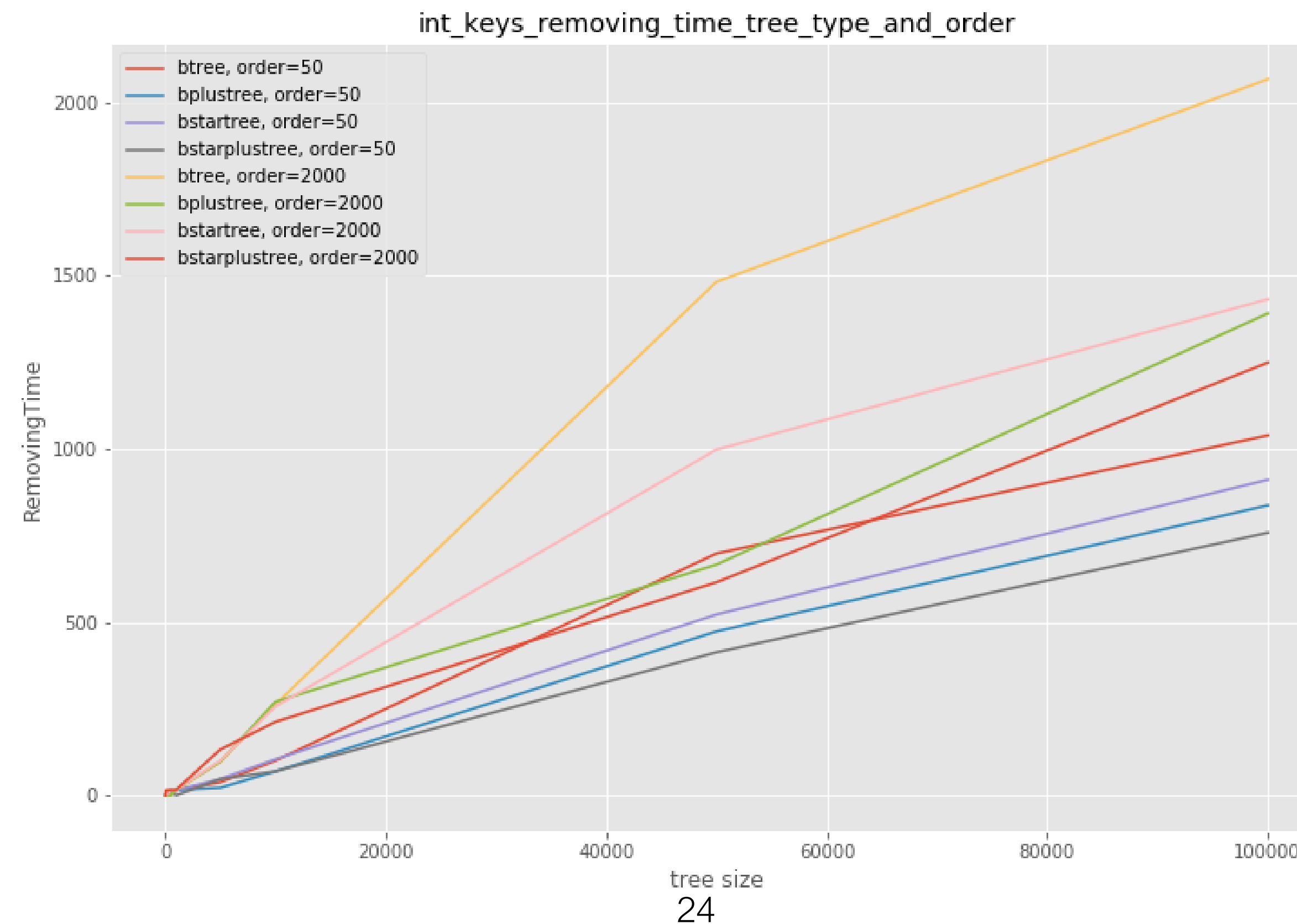
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Поиск в дереве (целочисленные ключи) – память и дисковые операции



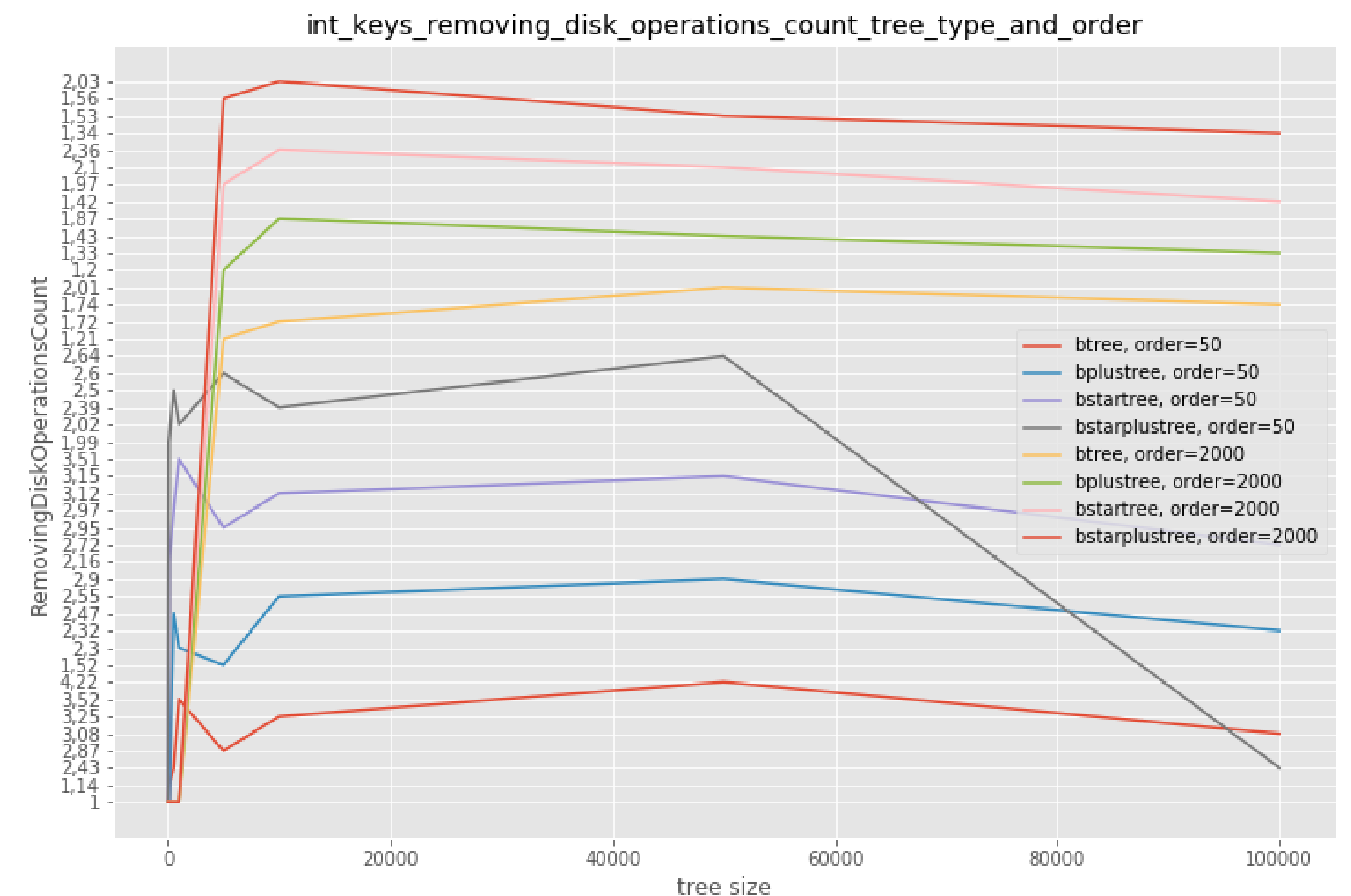
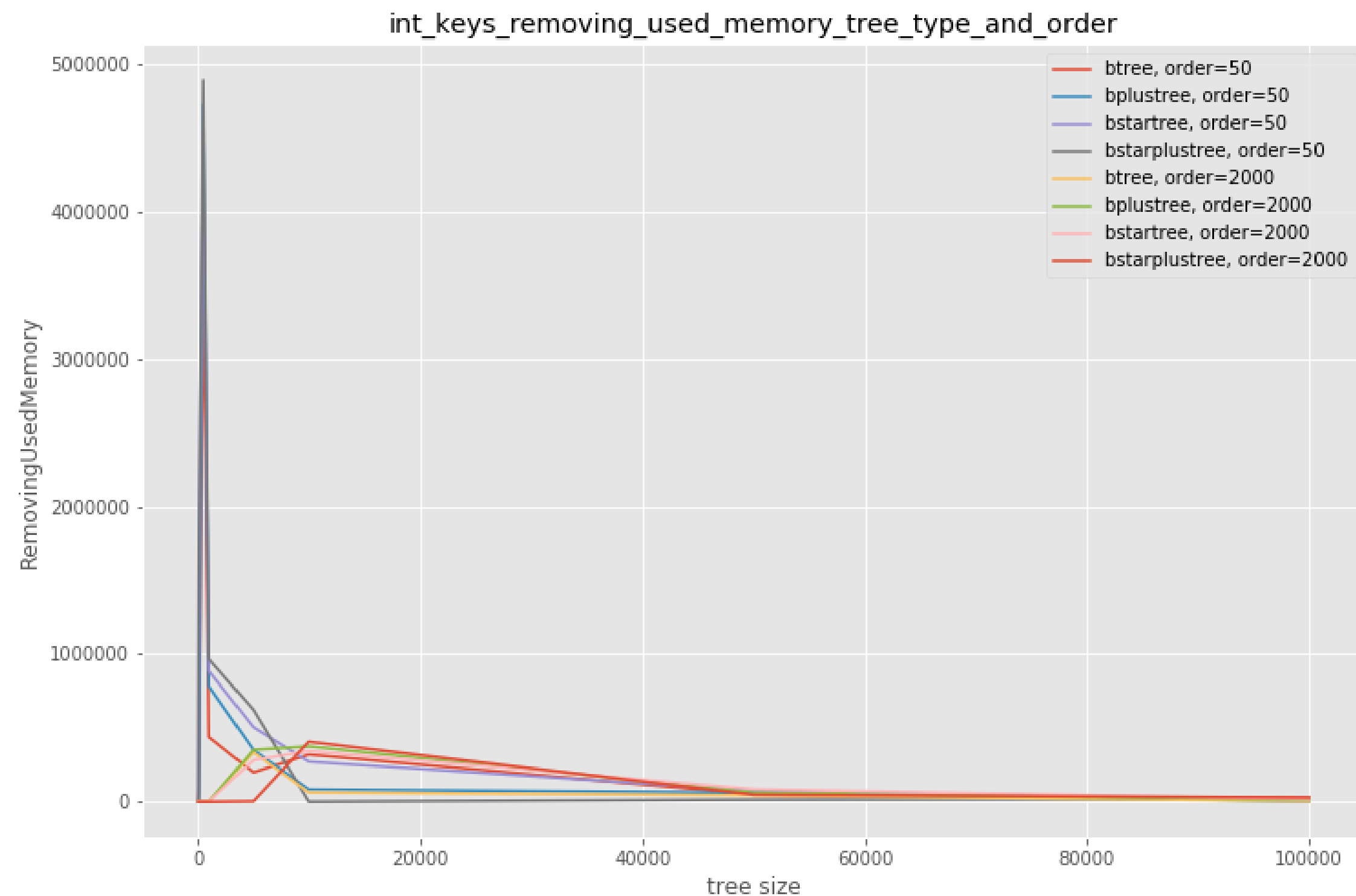
# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Удаление в дереве (целочисленные ключи) – время



# РАНЕЕ ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Удаление в дереве (целочисленные ключи) – память и дисковые операции



# АЛГОРИТМ ВЫБОРА СТРУКТУРЫ ДАННЫХ ДЛЯ ИНДЕКСАЦИИ

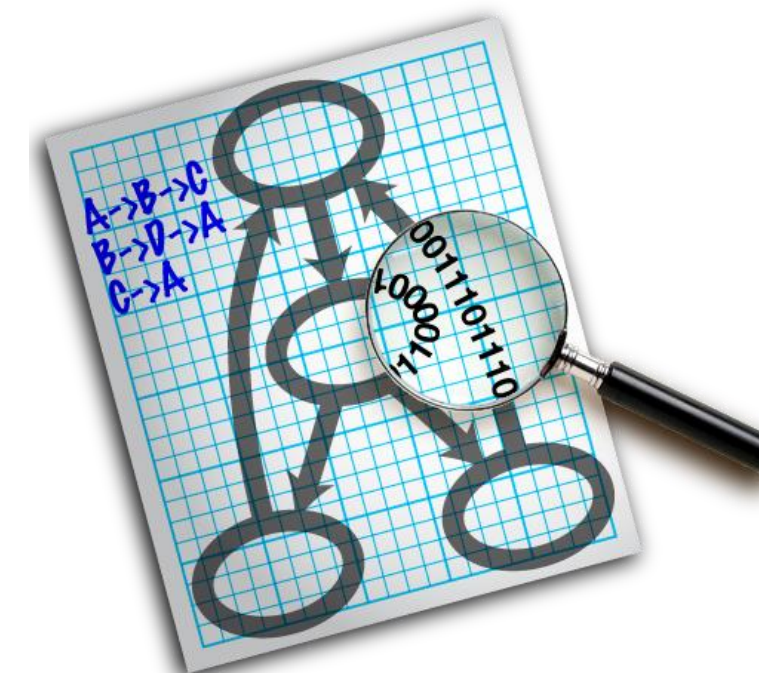
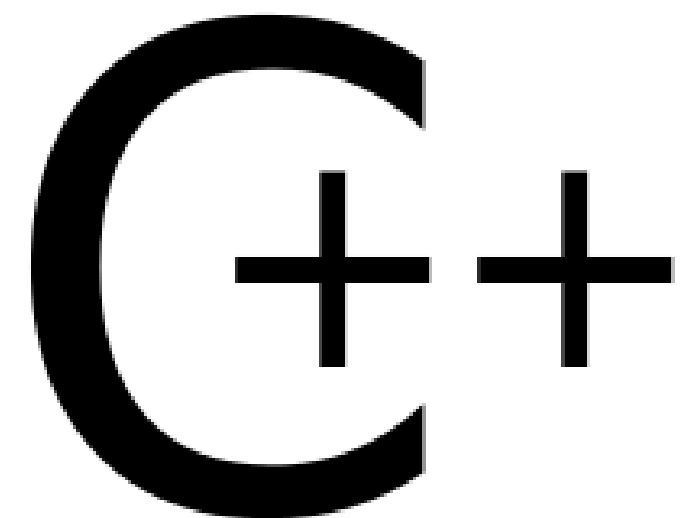
- Использует мультиклассовую линейную классификацию:

$$y_i = w_i^1 rows\_count + w_i^2 columns\_count + w_i^3 index\_size + \beta_i, \\ i = 1, 2, 3, 4$$

- Дерево, соответствующее  $i$ -му линейному классификатору выбирается как структура данных для индексирования таблицы тогда и только тогда, когда  $y_i = \max\{y_1, y_2, y_3, y_4\}$ , то есть  $i = \operatorname{argmax}_i(y_i)$
- Классификаторы обучаются с использованием Python 2 и библиотеки Sci-Kit Learn



# МЕТОДЫ И ТЕХНОЛОГИИ



## ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ

- Разработанное расширение для СУБД SQLite, позволяющее использовать в качестве индекса модификации В-дерева:  $B^+$ -дерево,  $B^*$ -дерево и  $B^{*+}$ -дерево
- Разработанный алгоритм выбора наиболее подходящей структуры данных (В-дерева либо одной из его модификаций) для индексирования таблицы
- Разработанное расширение для SQLite-менеджера для визуализации В-деревьев и их модификаций, а также рассчитанных для них метрик

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. B\*-tree [Электронный ресурс] // NIST Dictionary of Algorithms and Data Structures. Режим доступа: <https://xlinux.nist.gov/dads/HTML/bstartree.html>, свободный. (дата обращения: 23.12.2017)
2. B-Star Trees [Электронный ресурс] // College of Engineering, Computer Science & Construction Management, Chico, California State University. Режим доступа: [http://www.ecst.csuchico.edu/~mjstapleton/courses/Fall2007/CSCI311/ProgTwo\\_Bstar.htm](http://www.ecst.csuchico.edu/~mjstapleton/courses/Fall2007/CSCI311/ProgTwo_Bstar.htm), свободный. (дата обращения: 15.01.2018)
3. Kerttu Pollari-Malmi. B+-trees // [Электронный ресурс]: Computer Science | University of Helsinki. Режим доступа: <https://www.cs.helsinki.fi/u/mluukkai/tirak2010/B-tree.pdf>, свободный. (дата обращения: 07.12.2017)
4. Run-Time Loadable Extensions [Электронный ресурс] // SQLite. Режим доступа: <https://www.sqlite.org/loadext.html>, свободный (дата обращения: 04.11.2018)
5. Д. Кнут. Искусство программирования. Том 3. 2-е изд. — М.: ИД «Вильямс». — 2017. — 824 с.
6. Т. Кормен. Алгоритмы: построение и анализ. 3-е изд. / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. — М.: ИД «Вильямс». — 2013. — 1324 с.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

Спасибо за внимание!

[amrigin@edu.hse.ru](mailto:amrigin@edu.hse.ru)

[anton19979@yandex.ru](mailto:anton19979@yandex.ru)

[anton19979@yandex-team.ru](mailto:anton19979@yandex-team.ru)