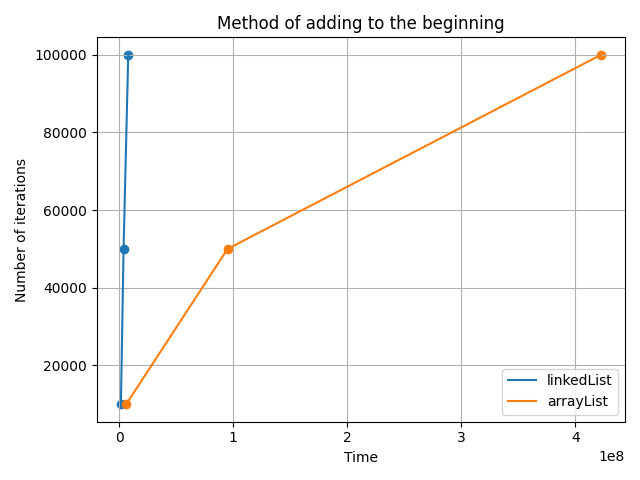
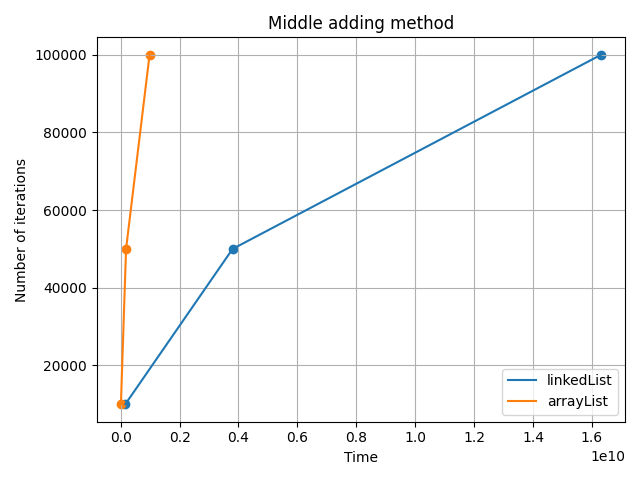
**Сравнение производительности коллекций**

1. ArrayList и LinkedList. Производилось три измерения для: 10.000, 50.000, 1.000.000 элементов (кроме удаления элементов из конца).
   1. Добавление элементов:

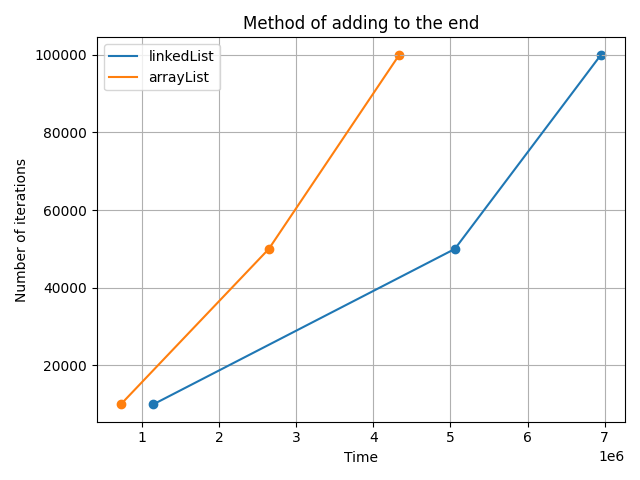
- в начало



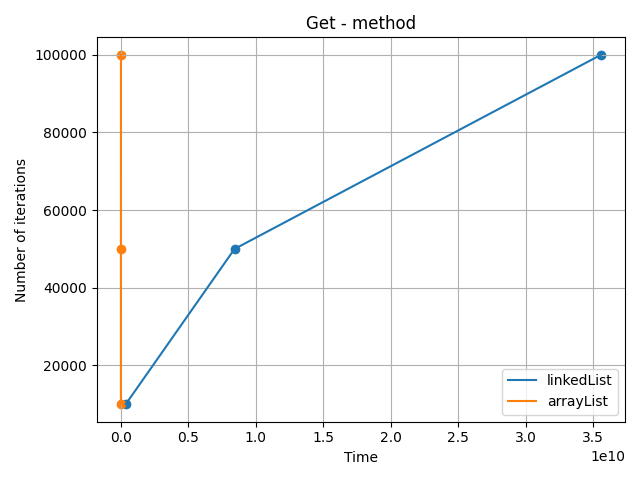
- в середину



- в конец

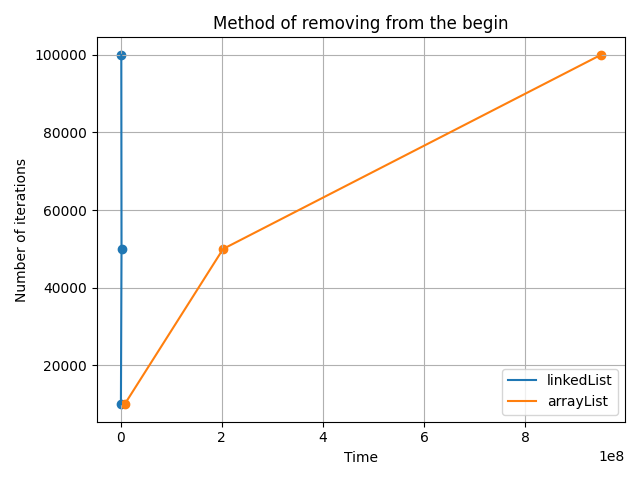


* 1. Получение элементов по индексу:

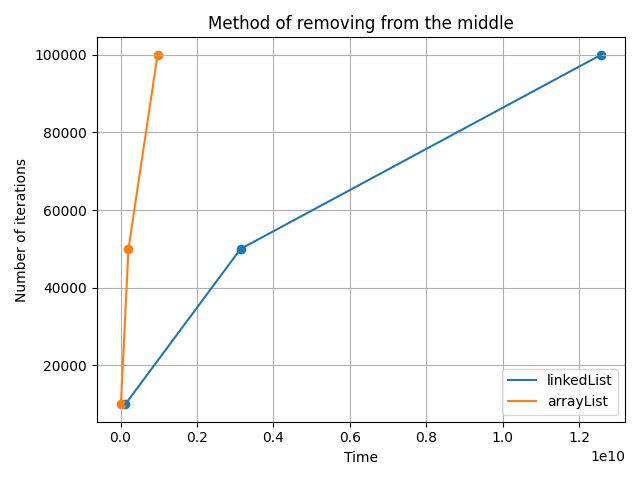


* 1. Удаление элементов:

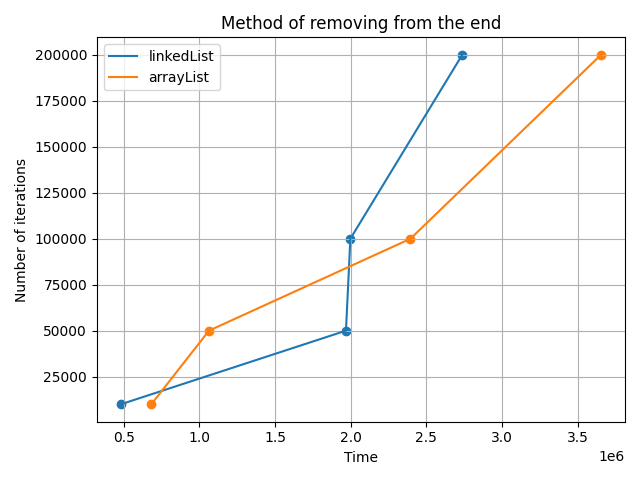
- из начала



- из середины



- из конца (здесь измерения проводились для: 10.000, 50.000, 100.000, 200.000)



**Вывод:**

Из графиков видно, что вставка в:

- начало быстрее происходит в LinkedList`е;

- середину – в ArrayList`е;

- конец – в ArrayListе.

Получение элементов по индексу быстрее в ArrayList`е.

Удаление элементов из:

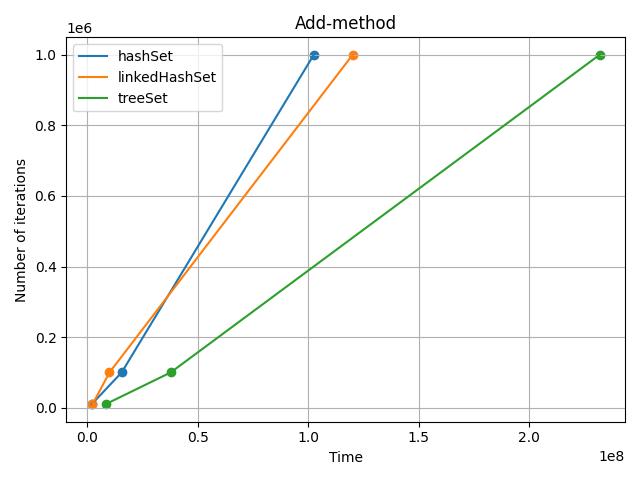
- начала быстрее происходит в LinkedList`е;

- середины – в ArrayList`е.

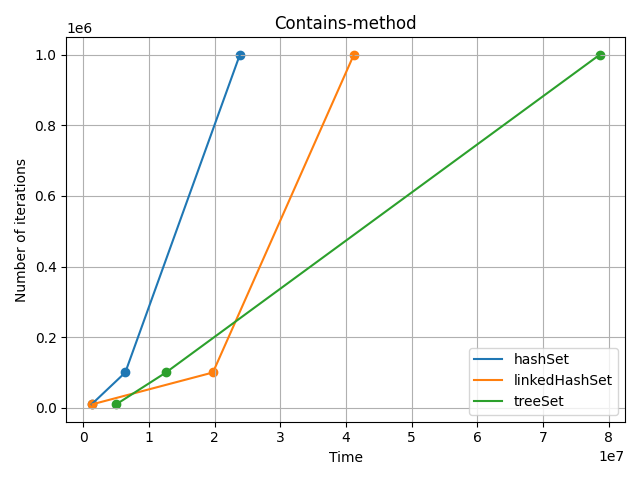
Удаление из конца производилось для: 10.000, 50.000, 100.000, 200.000 элементов. На графике видно, что, чем больше элементов, тем быстрее удаление происходит в LinkedList`е. При этом, если элементов немного (10.000-50.000), то удаление эффективнее в ArrayList`е.

Полученные здесь результаты объясняются устройством коллекций. ArrayList – это список, основанный на массиве. Тогда как LinkedList – связный список: элементы здесь, помимо хранения значения самого элемента, имеют ссылки на предыдущий и следующий элементы. В связи с этим, если необходима работа с началом и концом, то используют LinkedList. Тогда как, если требуется работа с элементами по индексу, то используют ArrayList.

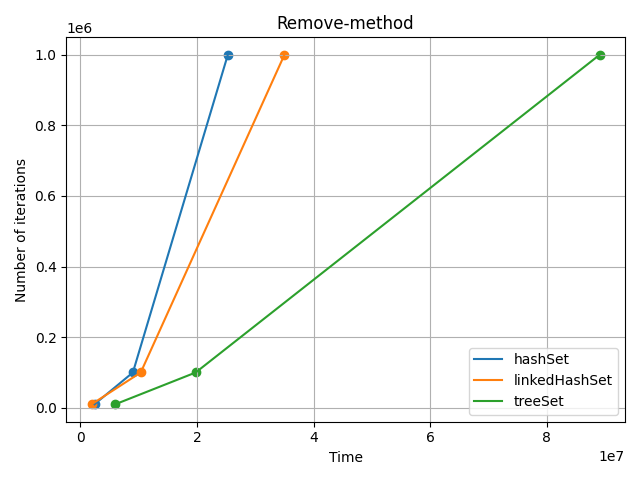
1. HashSet, LinkedHashSet, TreeSet. Производилось три измерения для: 10.000, 100.000, 1.000.000 элементов.
   1. Добавление элементов:



* 1. Поиск элементов:



* 1. Удаление элементов:



**Вывод:**

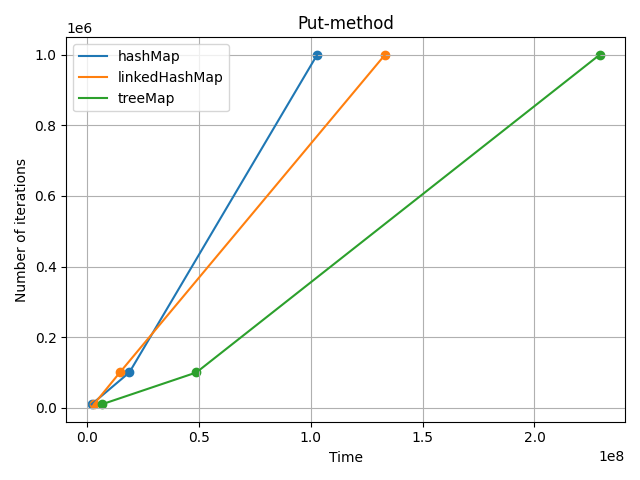
Во всех случаях медленнее оказался TreeSet. Самым быстрым – HashSet. LinkedHashSet не сильно отстает от HashSet`а. Связано это с их устройством:

- HashSet используется там, где порядок элементов не имеет значения;

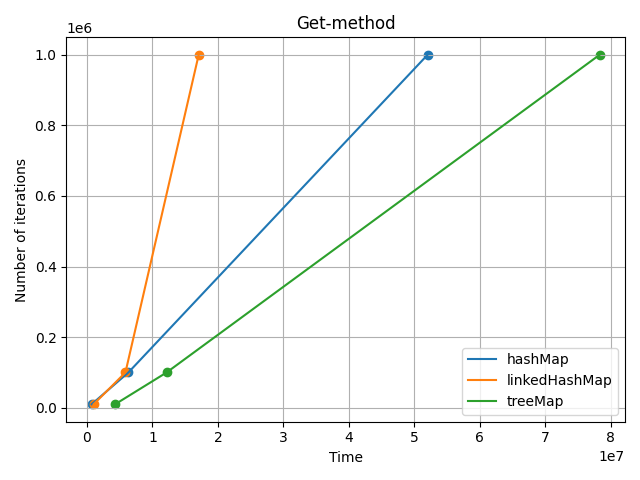
- LinkedHashSet хранит элементы в порядке добавления;

- TreeSet для хранения элементов применяет дерево. Элементы здесь располагаются в отсортированном по возрастанию виде.

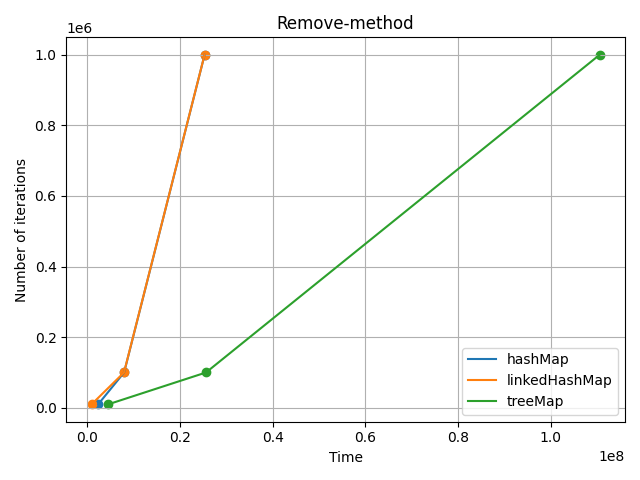
1. HashMap, LinkedHashMap, TreeMap. Производилось три измерения для: 10.000, 100.000, 1.000.000 элементов.
   1. Добавление элементов:



* 1. Получение элемента по ключу:



* 1. Удаление элемента по ключу:



**Вывод:**

Во всех случаях медленнее оказался TreeMap. Самым быстрым для добавления в случае большого количества элементов является – HashMap. В случае получения элементов (также при их большом количестве) быстрее оказывается LinkedHashMap. При удалении элементов HashMap и LinkedHashMap почти не отличаются. Связано это с их устройством.