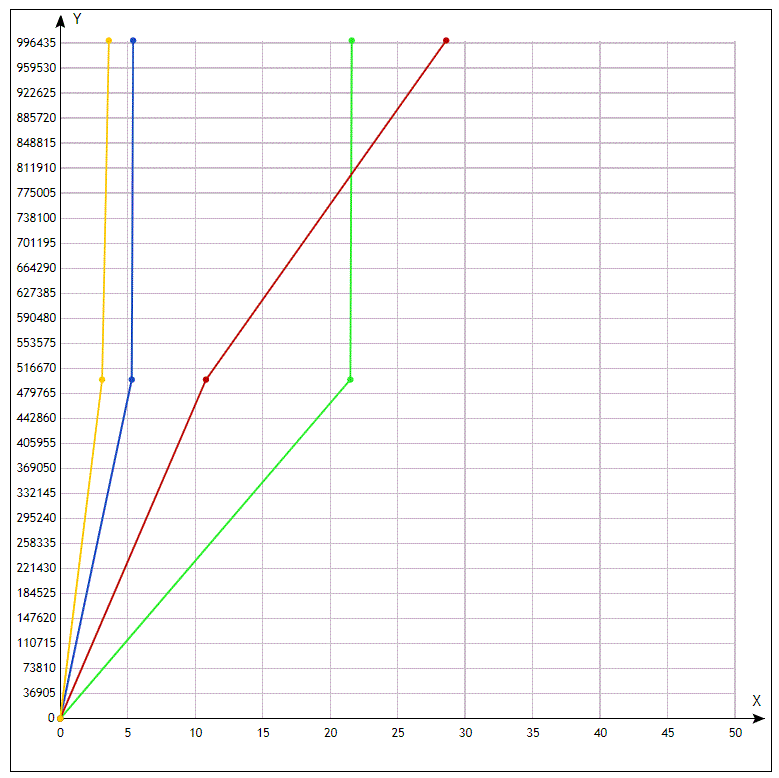
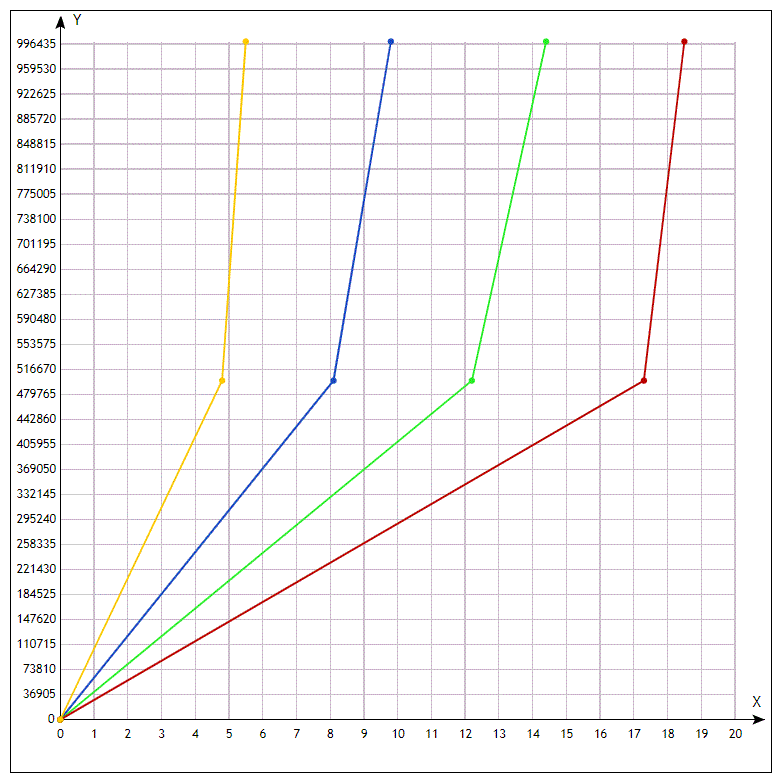
Производительность различных коллекций

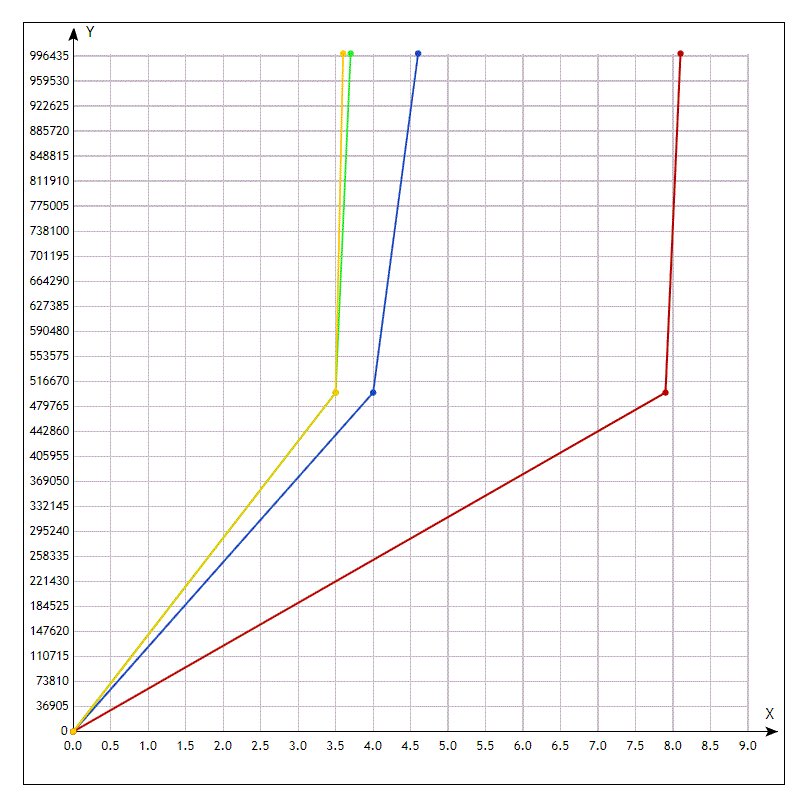
метод добавление ( жёлтый - ArrayList, синий - LinkedList, бордовый - TreeSet, зелёный HasSet)



Метод - удаление ( жёлтый - ArrayList, синий - LinkedList, бордовый - TreeSet, зелёный HasSet)



метод поиск ( жёлтый - ArrayList, синий - LinkedList, бордовый - TreeSet, зелёный HasSet)



Вывод: самым медленным оказался TreeSet, остальные коллекции проявляют себя быстрее или медленнее в зависимости от случая.

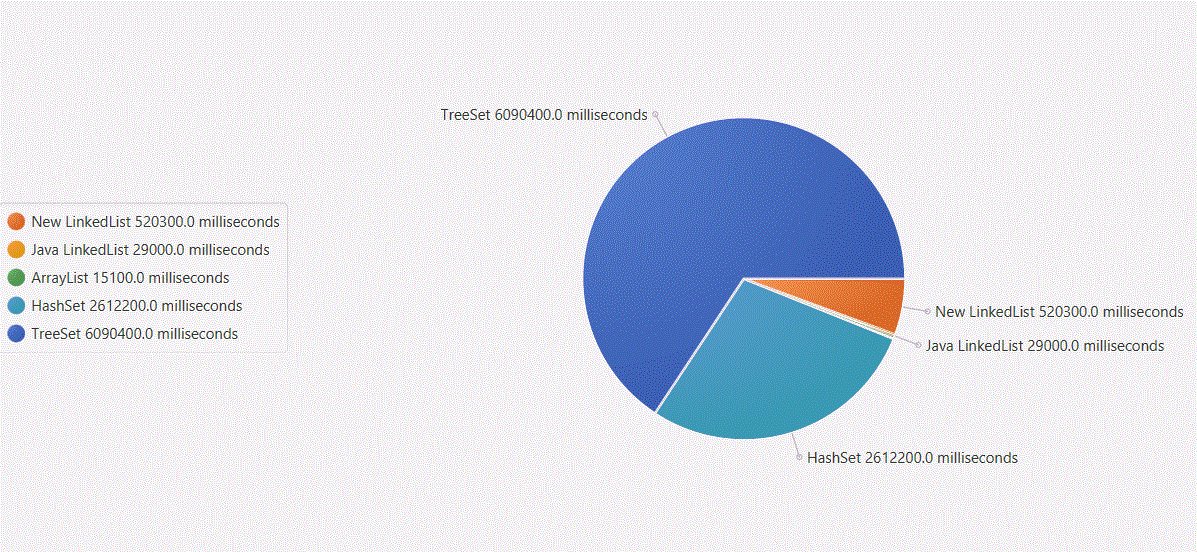
TreeSet - устроен на основе дерева и имеет в лучшем случае во всех операциях 0(logN) и при добавление автоматически сортирует элементы, но при добавлении одинаковых элементов, временная сложность поиска получилась хуже чем у LinkedList/ArrayList, при добавлении различных элементов коллекция работала быстрее.

ArrayList - расширяемый массив, поэтому имеет преимущество только в поиске элемента по индексу О(1), в других случаях имеет временную сложность О(n).

LinkedList - устроен на основе двусвязного списка и имеет сложность О(n) в операциях поиска по индексу и по значению, так как информация о следующем элементе списка находится у предыдущего и наоборот (список двусвязный), но при добавлении элемента или его удалении имеет сложность O(1) так как не требует перемещения элементов как ArrayList.

HashSet в тестах получился не самым быстрым, скорее всего это связанно с тем, что в тестировании производительности я заполнил коллекцию одинаковыми элементами и потом тестировал коллекцию точно таким же элементом , из за этого сложность поиска, которая может быть равна О(1) свелась примерно к такой же как у ArrayList/LinkedList или даже хуже, при добавлении разных элементов, коллекция работала быстрее всего.

Сравнение скорости добавления множества различных элементов в коллекции



Самыми медленными оказались TreeSet и HashSet, это связанно с тем, что TreeSet перед добавлением автоматически сортирует элементы и на это затрачивается время и ресурсы hashset, так же сортирует элементы по корзинам и так же тратит на это ресурсы.