Отчет по домашнему заданию Сортировка чисел

Цель:

Потренироваться в применении стандартных коллекций.

Описание/Пошаговая инструкция выполнения домашнего задания:

Реализовать сортировку массива чисел используя любой алгоритм из представленных на занятии (или свой, но с комментариями).

Отчет:

В качестве сравнения времени сортировки были выбраны сортировка пузырьком и стандартная библиотека Java java.util.Collections.sort().

Bubble sort (сортировка пузырьком) – это один из самых простых алгоритмов сортировки, данный алгоритм меняет местами два соседних элемента, если первый элемент массива больше второго. Так происходит до тех пор, пока алгоритм не обменяет местами все неотсортированные элементы. Сложность данного алгоритма сортировки равна O(n^2). Это означает, что время выполнения этого алгоритма увеличивается быстро с увеличением размера входных данных, поэтому он не подходит для работы с большими объемами данных.

Collections.sort() - это метод Java для сортировки коллекций, использует алгоритм QuickSort. Она имеет алгоритмическую сложность O (n log n). Он обеспечивает более быструю сортировку, чем сортировка пузырьком, особенно при работе с большими объемами данных.

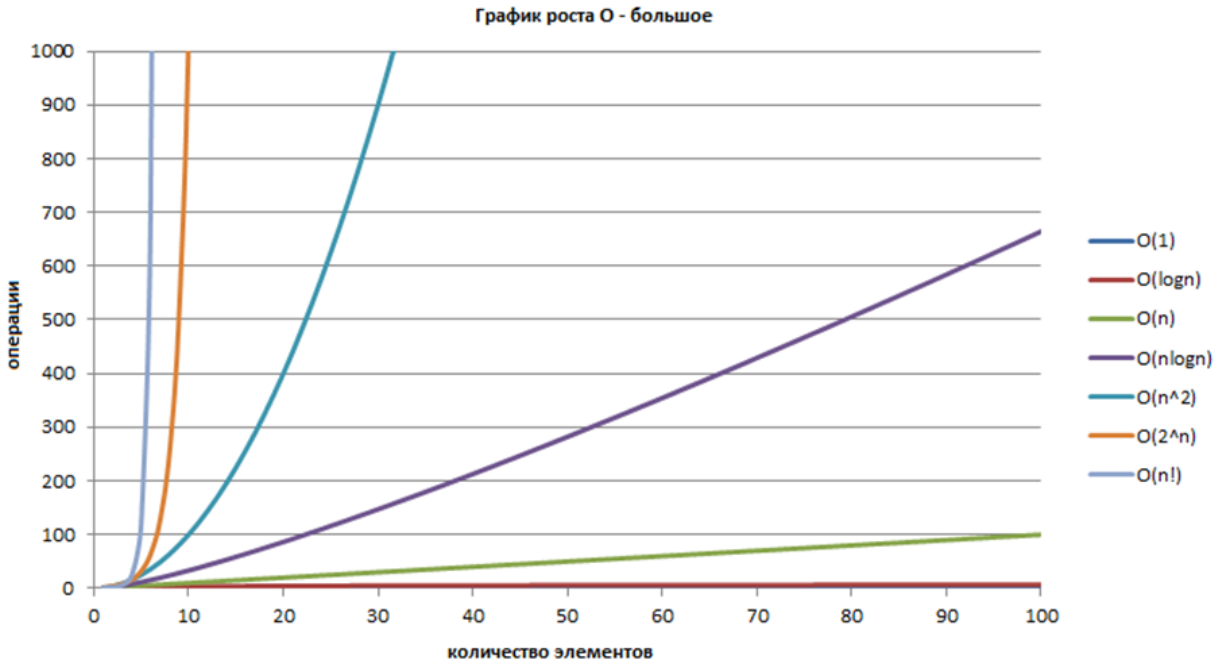
Collections.sort() использует адаптивную версию QuickSort, которая автоматически переключается на другой алгоритм сортировки (например, MergeSort) в случае, если размер входных данных меньше определенного порогового значения. Это повышает эффективность алгоритма и уменьшает вероятность его замедления. Это удобно и экономит время, вместо того, чтобы реализовывать его самостоятельно.

Но иногда ситуации бывают разные и может случиться, что сортировка пузырьком может оказаться удобнее и проще для решения конкретных задач. Однако в целом, Collections.sort() является более предпочтительным выбором для сортировки больших объемов данных в Java.

```java
    int[] array = new int[100000];
    Random rand = new Random();
    for (int i = 0; i < array.length; i++) {
        array[i] = rand.nextInt( bound: 10000);
    }
    long start = System.currentTimeMillis();

    bubbleSort(array);

    long end = System.currentTimeMillis();
    System.out.println("Время сортировки пузырьком: " + (end - start) + " мс");

    ArrayList<Integer> arrayList = new ArrayList<>();
    for (int i = 0; i < 100000; i++) {
        arrayList.add(rand.nextInt( bound: 10000));
    }

    start = System.currentTimeMillis();
    Collections.sort(arrayList);
    end = System.currentTimeMillis();
    System.out.println("Время сортировки методом Collections.sort(): " + (end - start) + " мс");
}
```

```
CompareSort × CompareSort ×
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Com
Время сортировки пузырьком: 36649 мс
Время сортировки методом Collections.sort(): 136 мс
```

В качестве эксперимента, на вход получено 10_000 случайных чисел из 100_000_000 чисел. Сортировка с помощью collection.sort быстрее чем сортировка пузырьком в 269 раз. Ниже приведены графики роста при различных сложностях.



Таким образом**, Collections.sort() является более быстрым и эффективным алгоритмом, чем сортировка пузырьком, особенно при больших объемах данных.**