Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учереждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно-

вычислительных систем (КИБЭВС)

ОТЧЕТ

по Лабораторной работе №3

«Сортировки»

Выполнил студент гр. 745

\_\_\_\_\_\_\_\_ Чубаров А.Н.

\_\_\_\_\_\_\_\_Фельчук М.В.

\_\_\_.\_\_\_.2017

Доцент кафедры БИС

\_\_\_\_\_\_\_\_\_Романов А.С.

\_\_\_.\_\_\_.2017

**1 Введение**

**Цель работы:** Научиться применять методы сортировки.

**2 Ход работы**

3 вариант

2.1 Shell sort

Сортировка Шелла ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Shell sort) — [алгоритм сортировки](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B8), являющийся усовершенствованным вариантом [сортировки вставками](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%B2%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0%D0%BC%D0%B8). Идея метода [Дональда Шелла](https://ru.wikipedia.org/wiki/%D0%A8%D0%B5%D0%BB%D0%BB,_%D0%94%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%B4) состоит в сравнении элементов, стоящих не только рядом, но и на определённом расстоянии друг от друга; иными словами — это сортировка вставками, но с предварительными «грубыми» проходами.

****

Рисунок 2.1 - Код программы для Shell sort

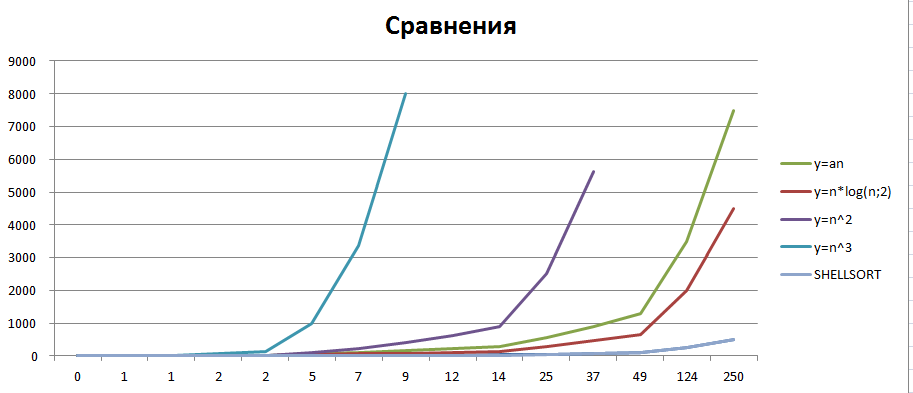


Рисунок 2.2 - Сравнения для ShellSort



Рисунок 2.3 - Перестановки для ShellSort

При сортировке Шелла сначала сравниваются и сортируются между собой значения, стоящие один от другого на некотором расстоянии {\displaystyle d}. После этого процедура повторяется для некоторых меньших значений {\displaystyle d}, а завершается сортировка Шелла упорядочиванием элементов при {\displaystyle d=1} (то есть обычной [сортировкой вставками](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%B2%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%BE%D0%B9)). Эффективность сортировки Шелла в определённых случаях обеспечивается тем, что элементы «быстрее» встают на свои места (в простых методах сортировки, например, [пузырьковой](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%BE%D0%BC_%D0%BF%D1%83%D0%B7%D1%8B%D1%80%D1%8C%D0%BA%D0%B0), каждая перестановка двух элементов уменьшает количество [инверсий](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F_(%D0%BF%D0%B5%D1%80%D0%B5%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0)) в списке максимум на 1, а при сортировке Шелла это число может быть больше).

Невзирая на то, что сортировка Шелла во многих случаях медленнее, чем [быстрая сортировка](https://ru.wikipedia.org/wiki/%D0%91%D1%8B%D1%81%D1%82%D1%80%D0%B0%D1%8F_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0), она имеет ряд преимуществ:

* отсутствие потребности в памяти под стек;
* отсутствие деградации при неудачных наборах данных — быстрая сортировка легко деградирует до O(n²), что хуже, чем худшее гарантированное время для сортировки Шелла.



Рисунок 2.4 - Время для ShellSort

2.2 Selection sort

Сортировка выбором (Selection sort) — [алгоритм сортировки](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B8). Может быть как устойчивый, так и неустойчивый. На массиве из n элементов имеет время выполнения в худшем, среднем и лучшем случае [Θ](https://ru.wikipedia.org/wiki/O-%D0%BD%D0%BE%D1%82%D0%B0%D1%86%D0%B8%D1%8F)(n2), предполагая что сравнения делаются за постоянное время.

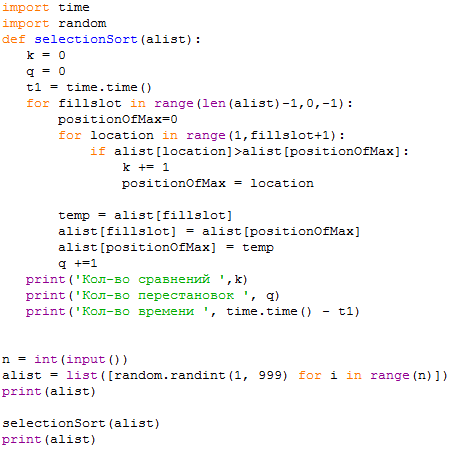
****

Рисунок 2.5 - Код программы для Selection sort

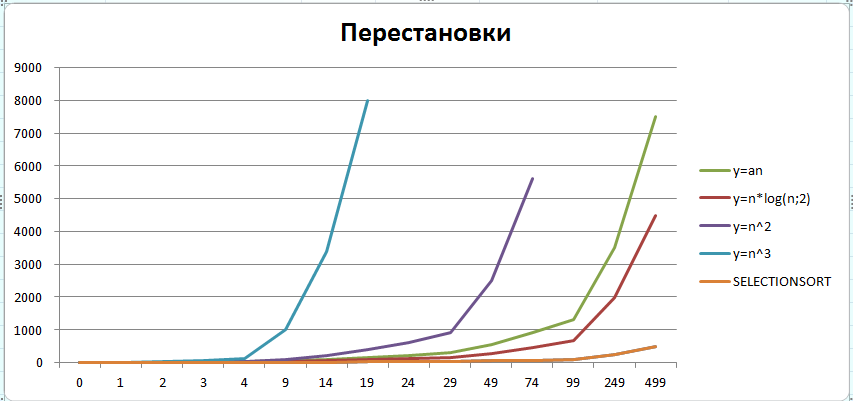


Рисунок 2.6 - Перестановки для Selection Sort

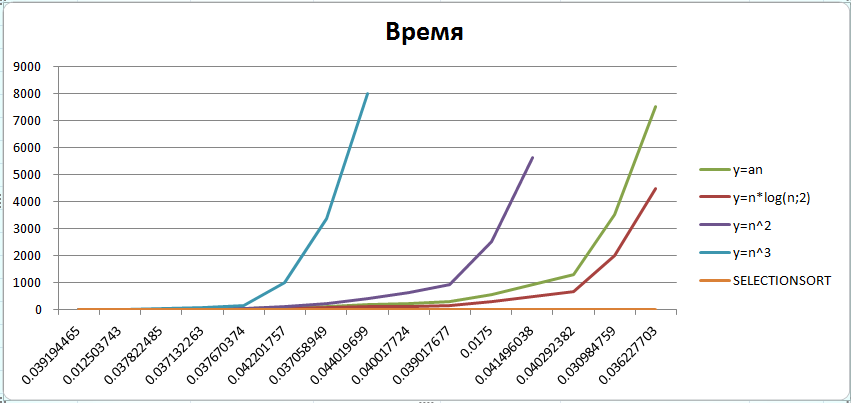


Рисунок 2.7 - Время для Selection Sort



Рисунок 2.8 - Сравнения для Selection Sort

**Наихудший случай:**

Число сравнений в теле цикла равно (N-1)\*N/2.

Число сравнений в заголовках циклов (N-1)\*N/2.

Число сравнений перед операцией обмена N-1.

Суммарное число сравнений N2−1.

Число обменов N-1.

**Наилучший случай:**

Время сортировки 10000 коротких целых чисел на одном и том же [программно-аппаратном комплексе](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE-%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BB%D0%B5%D0%BA%D1%81) сортировкой выбором составило ≈40сек., а ещё более улучшенной [сортировкой пузырьком](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BF%D1%83%D0%B7%D1%8B%D1%80%D1%8C%D0%BA%D0%BE%D0%BC) ≈30сек.

[Пирамидальная сортировка](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%80%D0%B0%D0%BC%D0%B8%D0%B4%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0) сильно улучшает базовый алгоритм, используя структуру данных «[куча](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D1%83%D1%8E%D1%89%D0%B5%D0%B5_%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE)» для ускорения нахождения и удаления минимального элемента.

Существует также двунаправленный вариант сортировки методом выбора, в котором на каждом проходе отыскиваются и устанавливаются на свои места и минимальное, и максимальное значения.

2.3 Heap sort

Основная идея — ищем максимальный элемент в неотсортированной части массива и ставим его в конец этого подмассива. В поисках максимума подмассив перестраивается в так называемое **сортирующее дерево** (она же **двоичная куча**, она же **пирамида**), в результате чего максимум сам «всплывает» в начало массива. После этого перемещаем максимум в конец подмассива. Затем над оставшейся частью массива снова осуществляется процедура перестройки в сортирующее дерево с последующим перемещением максимума в конец подмассива.

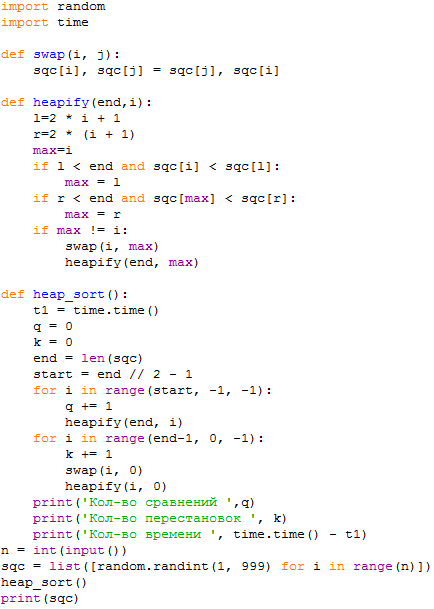


Рисунок 2.9 - Код программы для Heap sort

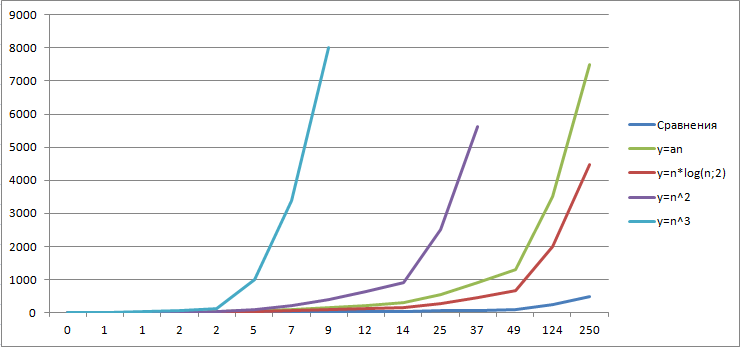


Рисунок 2.10 - График для характеристики сравнения

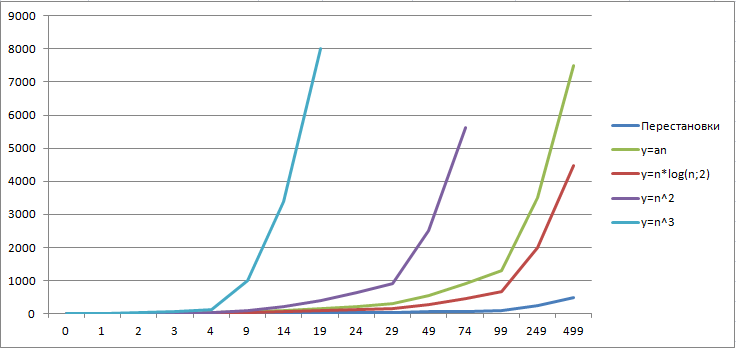


Рисунок 2.11 - График для характеристики перестановки

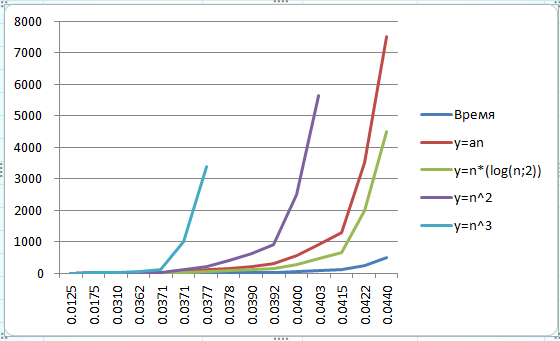


Рисунок 2.12 - График для характеристики времени

Построение пирамиды занимает O(n log n) операций, причем более точная оценка дает даже O(n) за счет того, что реальное время выполнения downheap зависит от высоты уже созданной части пирамиды.

Вторая фаза занимает O(n log n) времени: O(n) раз берется максимум и происходит просеивание бывшего последнего элемента. Плюсом является стабильность метода: среднее число пересылок (n log n)/2, и отклонения от этого значения сравнительно малы.

Пирамидальная сортировка не использует дополнительной памяти.

Метод не является устойчивым: по ходу работы массив так "перетряхивается", что исходный порядок элементов может измениться случайным образом.

Поведение неестественно: частичная упорядоченность массива никак не учитывается.

**3 Заключение**

Мы научись применять методы сортировки.