Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им.И.Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление:710400 «Программная инженерия»

ОТЧЕТ

По дисциплине: «Алгоритмы и структуры данных»

Лабораторная работа №10

Тема: «Анализ улучшенных сортировок»

Выполнил: студент группы

ПИ(б)-2-19 Улан уулу Нурдин

Проверила: Валеева А. А.

Бишкек – 2020

**Практическое задание**

**I. Ответы на контрольные вопросы**

**1. В чем заключается особенность алгоритма Шелла? Опишите суть алгоритма.**

В данном методе сначала сортируются элементы, которые стоят на расстоянии n/2 элементов друг от друга, затем элемент стоящие на расстоянии n/4, затем n/8 и т.д. пока n/m не будет равно 1, затем осуществляется сортировка при помощи сортировки вставками.

**2. В чем заключается особенность алгоритма быстрой сортировки? Опишите суть алгоритма.**

Для начала указываем опорный элемент. Проверяем все элементы, стоящие левее опорного, если встречается элемент, стоящий левее опорного, но больше по значению то запоминается его индекс(i), затем, проверяются все элементы правее опорного, если встречается значение меньше опорного, то запоминается его индекс(j). Затем производится обмен позиций элементов на позициях i и j. Данное действие выполняется пока i меньше j.

**3. В чем заключается алгоритм пирамидальной сортировки? Опишите суть метода.**

Для начала раскладываем массив в виде бинарного дерева, arr[0] становится корнем дерева, а следующие элементы потомками.

Далее проверяем самую нижнюю вершину с его потомками и ищем минимальный, и поднимаем на верх, проворачиваем эту махинацию для каждой вершины, пока не дойдем до корня. Далее новый корень, который самый минимальный элемент переносим в новый массив в начало, и делаем так пока, не закончатся элементы в бинарном дереве.

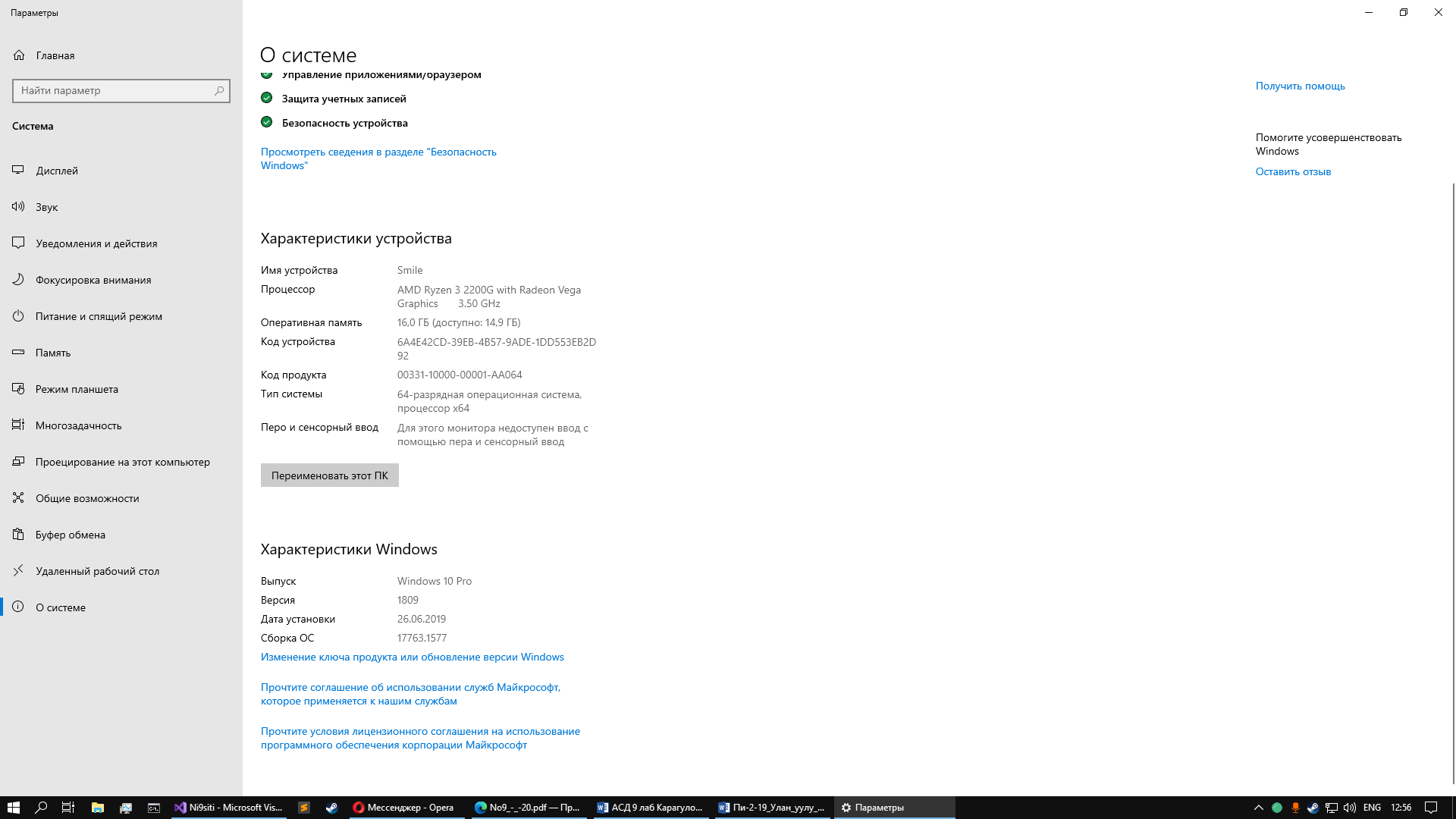
**4. В чем заключается алгоритм сортировки слиянием? Опишите суть метода.**

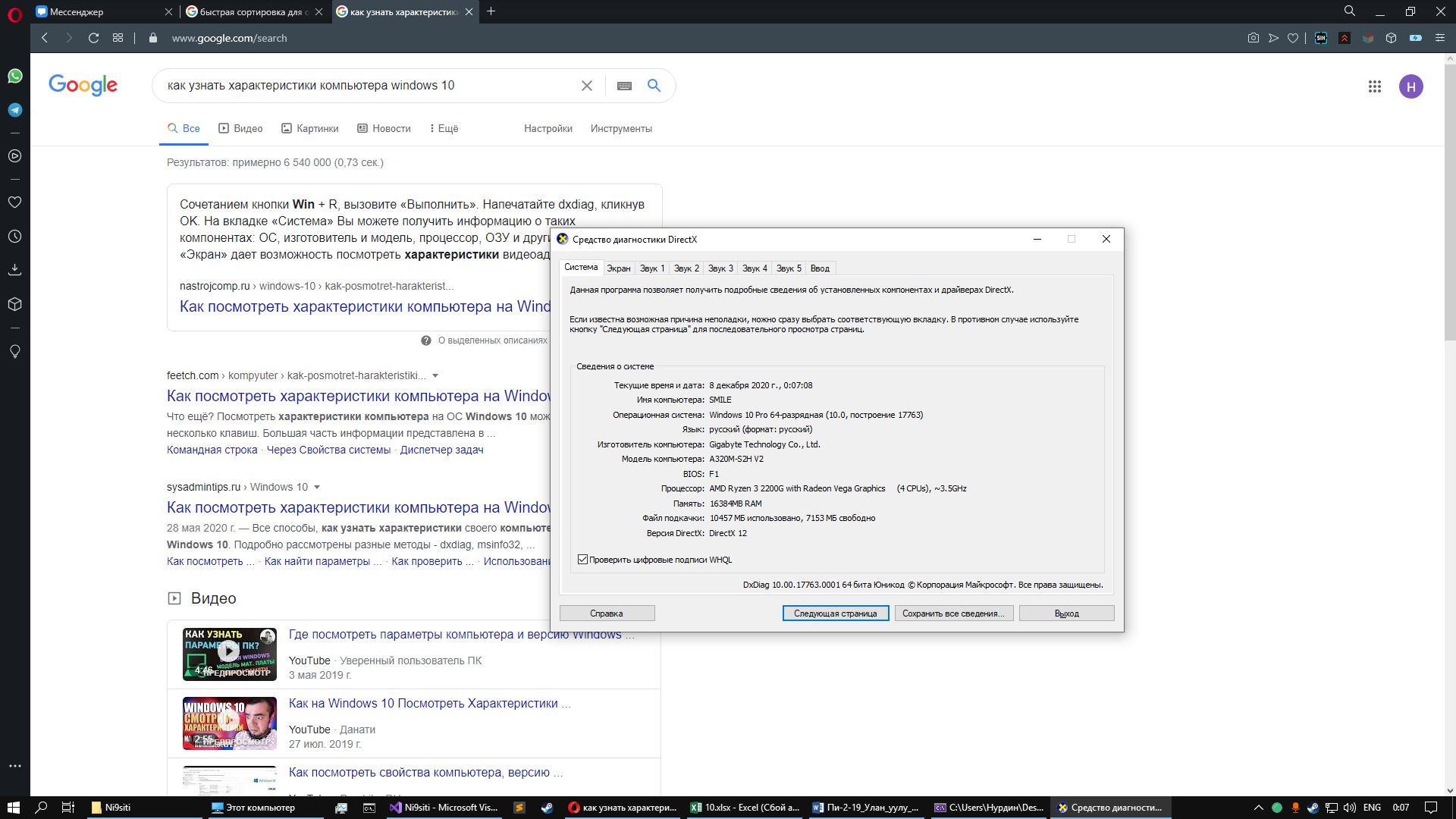
Для начала нам нужно определить точку выхода из рекурсии, то есть n < 2. Далее разделить массив на два под-массива, и сортировать левую сторону, потом правую, после сливать сортируя отсортированные стороны. И так продолжая до того момента пока массив не будет весь отсортирован.

**II. Выполнить задания**

Выполнить исследование улучшенных методов сортировки (алгоритм Шелла, алгоритм быстрой сортировки, алгоритм пирамидальной сортировки, radix) для числового массива и связанного списка (по тем же данным) по времени выполнения и указанным параметрам. (мощность массива n=10, 50, 100, 1000, 10000,100000).

**Указать характеристики ПК:**





**Замеры времени выполнения производятся при помощи функции clock();**

**1.** Провести анализ проведенных расчетов для лучшего, среднего и лучшего случаев для всех улучшенных алгоритмов сортировки массива. Для всех видов сортировок, для структур данных (массив, статический список односвязный и двусвязный список) использовать одни и те же приготовленные заранее данные, записанные в файлы. Численный эксперимент проводить для лучшего случая (данные отсортированы в заданном порядке), для среднего случая (случайным образом сформированы данные), для худшего случая (данные отсортированы в обратном порядке).

Для мощностей - N - 10, 50 - диапазон случайных данных- -100 до 100;

N - 100: -1000 до 1000;

N - 1000: -10000 до 10000;

N - 10000: -100000 до 100000;

N -100000: -1000000 до 1000000.

**2.** Провести анализ проведенных расчетов для лучшего, среднего и лучшего случаев для всех улучшенных алгоритмов сортировки односвязного списка. (для тех же данных и параметров). Параметр сложности: количество операций сравнений; количество операций перестановок. Показать зависимость от мощности данных. Обоснованный вывод по результатам анализа.

**3.** Провести анализ проведенных расчетов для лучшего, среднего и лучшего случаев для всех улучшенных алгоритмов сортировки двусвязного списка (для тех же данных и параметров). Параметр сложности: количество операций сравнений; количество операций перестановок. Показать зависимость от мощности данных. Обоснованный вывод по результатам анализа. Результаты анализа представить в виде таблицы Excel и графиков.

**Лучший случай для массива:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1 000 | 10 000 | 100 000 | 1 000 000 | 10 000 000 | 100 000 000 |
|
| Шелла | Время в Секунды | 0 | 0 | 0 | 0 | 0,1 | 0,8 | 0,95 | 1,1 | 13,7 |
| Кол-во сравнений | 22 | 203 | 503 | 8006 | 120005 | 1500006 | 18000007 | 220000008 | 2500000012 |
| Кол-во перестановок | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Быстрая | Время в Секунды | 0 | 0 | 0 | 0,1 | 0,2 | 0,25 | 0,39 | 7,8 | 81,3 |
| Кол-во сравнений | 23 | 208 | 501 | 8173 | 114321 | 1461796 | 16540781 | 183150114 | 1999918885 |
| Кол-во перестановок | 7 | 35 | 67 | 674 | 9558 | 126563 | 2510764 | 40799412 | 581845603 |
| Пирамидальная | Время в Секунды | 0 | 0 | 0 | 0,2 | 0,22 | 0,28 | 3,1 | 38,6 | 424,1 |
| Кол-во сравнений | 77 | 836 | 2122 | 34545 | 478301 | 6112074 | 70053470 | 762410371 | 8686762329 |
| Кол-во перестановок | 31 | 268 | 644 | 9683 | 131261 | 1648036 | 18803860 | 205744319 | 2334981287 |

**Худший случай для массива:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1 000 | 10 000 | 100 000 | 1 000 000 | 10 000 000 | 100 000 000 |
|
| Шелла | Время в Секунды | 0 | 0 | 0 | 0 | 0,1 | 0,13 | 0,2 | 1,6 | 17,4 |
| Кол-во сравнений | 27 | 261 | 668 | 11697 | 170564 | 2173568 | 24898879 | 289801654 | 3201513846 |
| Кол-во перестановок | 13 | 103 | 260 | 4681 | 60546 | 773542 | 7898832 | 79801362 | 801510838 |
| Быстрая | Время в Секунды | 0 | 0 | 0 | 0,1 | 0,2 | 0,32 | 0,6 | 6,9 | 90,9 |
| Кол-во сравнений | 21 | 184 | 454 | 7682 | 109333 | 1411807 | 16040818 | 178150406 | 1949920109 |
| Кол-во перестановок | 12 | 58 | 117 | 1174 | 14558 | 176562 | 3010741 | 45799237 | 631843785 |
| Пирамидальная | Время в Секунды | 0 | 0 | 0 | 0,1 | 0,19 | 0,29 | 4,8 | 39,5 | 433,8 |
| Кол-во сравнений | 42 | 584 | 1519 | 28004 | 413570 | 5446654 | 67611888 | 806598733 | 9379640937 |
| Кол-во перестановок | 22 | 208 | 511 | 8316 | 116616 | 1497440 | 18300014 | 216075505 | 2491950673 |

**Средний случай для массива:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1 000 | 10 000 | 100 000 | 1 000 000 | 10 000 000 | 100 000 000 |
|
| Шелла | Время в Секунды | 0 | 0 | 0 | 0 | 0,2 | 0,29 | 0,46 | 6,4 | 84,9 |
| Кол-во сравнений | 26 | 327 | 801 | 14788 | 263088 | 4185324 | 66068472 | 1030210759 | 13951479112 |
| Кол-во перестановок | 9 | 149 | 346 | 7307 | 148277 | 2735992 | 48570989 | 815226173 | 11501520481 |
| Быстрая | Время в Секунды | 0 | 0 | 0 | 0 | 0,4 | 0,28 | 0,4 | 5,9 | 81,3 |
| Кол-во сравнений | 27 | 202 | 511 | 8679 | 116557 | 1540447 | 17917559 | 189000978 | 2064718381 |
| Кол-во перестановок | 9 | 69 | 117 | 1264 | 16795 | 191775 | 3244083 | 46971077 | 642296090 |
| Пирамидальная | Время в Секунды | 0 | 0 | 0 | 0,1 | 0,18 | 0,24 | 2,8 | 37,6 | 598,2 |
| Кол-во сравнений | 64 | 731 | 1830 | 31405 | 447036 | 5799275 | 71283933 | 845503010 | 9788046876 |
| Кол-во перестановок | 28 | 245 | 586 | 9081 | 124192 | 1574637 | 19046977 | 223827292 | 2571520270 |

**Сортировка Шелла**

Производительность в худшем случае O(n log2n), средняя производительность (зависит от последовательности пропусков), производительность в лучшем случае O(n log n). Это один из лучших вариантов, потому что он работает стабильно почти все время. Но сравнение с быстрой сортировкой сравнение элементов происходит чаще.

**Быстрая сортировка**

Наихудшая производительность O(n2), в наилучшей производительности O(nlogn), в средней производительности O(n log n). Это лучший алгоритм сортировки, так как он стабильно работает в любых случаях и имеет существенное преимущество перед другими рассмотренными методами.

**Пирамидальная сортировка**

Этот метод можно рассматривать как улучшенную сортировку выбора: подобно этому алгоритму, он делит свои входные данные на отсортированную и несортированную области и итеративно сокращает несортированную область, выделяя самый большой элемент и перемещая его в отсортированную область.

**Анализ улучшенных сортировок.**

По трем графикам видно то что, **пирамидальная** сортировка является наихудшей по времени, сравнениям и перестановкам. За ней идет **быстрая** сортировка, и самая быстрой оказалась сортировка **Шелла.**

Сортировка **Шелла** выигрывает из – за того что она не рекурсивна, и если какая - то часть массива отсортирована, то она не переставляет элементы как в **быстрой** или в **пирамидальной** сортировке.

Метод **Шелла** хорошо показывает себя в относительно небольших N, но уже в случае N = 100 000 000, его начинает опережать **быстрая** сортировка из - за количества сравнений и перестановок, так - как сортировка Шелла сравнивает в определенном шаге, а быстрая сортировка проверяет определенные под-массивы.

А пирамидальная сортировка проигрывает по всем пунктам, потому что даже в идеально отсортированном массиве, он сравнивает элемент вершины и элементы его потомков, и переставляет наибольшую вершину в отсортированную область массива.