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 №9

Тема: «Анализ простых 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. Что такое сортировка?**

Сортировка данных —это упорядочивание данных по возрастанию или убыванию. Сортировка производится по одному из ключей (значение и т.д.).

**2. Для каких задач применяется сортировка?**

В первую очередь, для поиска и представления данных. Некоторые задачи с неотсортированными данными решить очень трудно, а некоторые просто невозможно. Пример: орфографический словарь, в нем слова отсортированы по алфавиту.

**3. Какие виды сортировки существуют?**

**Простые**

Сортировка выбором

Сортировка вставками

Сортировка пузырьком

Сортировка шейкером

**Улучшенные**

Быстрая сортировка

Сортировка Шелла

**4. Назовите критерии эффективности алгоритмов сортировки.**

Существует 2 критерия это время выполнения и затраченная память, обычно чем меньше время, тем больше дополнительной памяти затрачивается.

**5. Инструменты оценки эффективности алгоритмов?**

Устойчивость — устойчивая сортировка не меняет взаимного расположения элементов с одинаковыми ключами.

Естественность поведения — эффективность метода при обработке уже упорядоченных или частично упорядоченных данных. Алгоритм ведёт себя естественно, если учитывает эту характеристику входной последовательности и работает лучше.

Потребности в дополнительной памяти или её отсутствию.

**6. Перечислите критерии выбора алгоритма сортировки.**

Скорость работы в среднем, лучшем, худшем случае и объём затраченной памяти.

**7. В чем заключается обменная сортировка? Опишите суть метода.**

Алгоритм основан на сравнении и перемене мест соседних элементов, если их ключи не упорядочены. Процесс продолжается до тех пор, пока не будут упорядочены все элементы исходной последовательности.

**8. В чем заключается сортировка выбором? Опишите суть метода.**

1. В неотсортированном части массива ищется локальный максимум (минимум).

2. Найденный максимум (минимум) меняется местами с последним (первым) элементом в неотсортированном части массива.

**9. В чем заключается сортировка вставками? Опишите суть метода.**

1. Перебираются элементы в неотсортированной части массива.

2. Каждый элемент вставляется в отсортированную часть массива на то место, где он должен находиться.

**10. В чем заключается шейкерная сортировка? Опишите суть метода.**

Это двунаправленная пузырьковая сортировка, суть в том, что мы за один раз проходим по массиву в одну сторону и другую.

**11. Какие факторы могут влиять на эффективность сортировки.**

Размер данных, большие размеры переставляются медленнее. Количество сортируемых элементов.

**II. Выполнить задания**

Выполнить исследование прямых методов сортировки (выбор, вставки, обмен, шейкерная сортировка) для числового массива (мощность массива n=10, 50, 100, 1000, 10000,100000) по времени выполнения и указанным параметрам. 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;

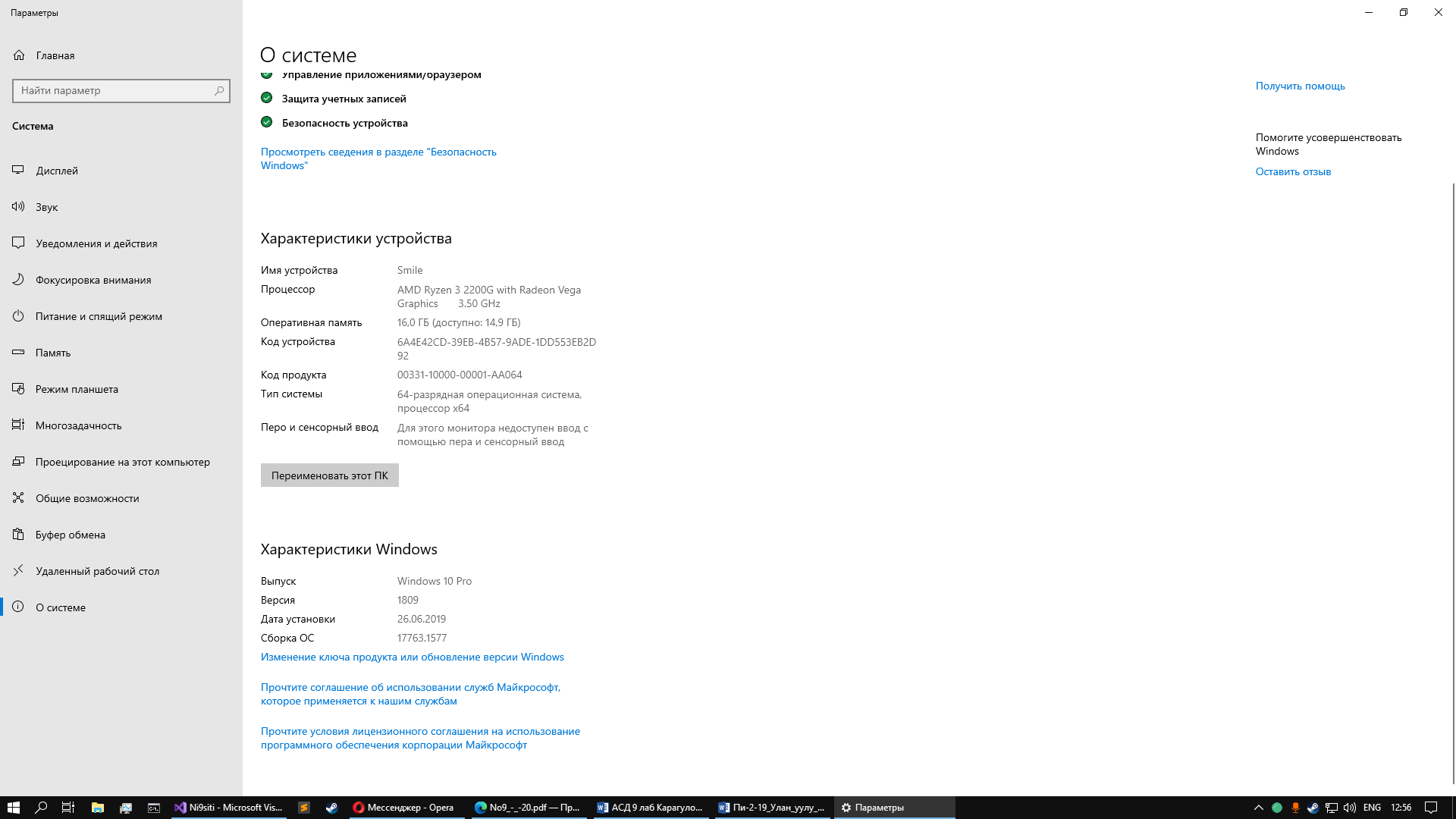
N - 100: -1000 до 1000;

N - 1000: -10000 до 10000;

N - 10000: -100000 до 100000;

N -100000: -1000000 до 1000000.

**Указать характеристики ПК:**



**2. Провести анализ проведенных расчетов для лучшего, среднего и лучшего случаев:**

метод выбора - параметр сложности: количество операций сравнений; количество операций перестановок. Показать зависимость от мощности данных.

метод вставки – параметр сложности: количество операций сравнений; количество операций перестановок. Показать зависимость от мощности данных.

метод обмена – параметр сложности: количество операций сравнений; количество операций перестановок. Показать зависимость от мощности данных.

шейкерная сортировка параметр сложности: параметр сложности: количество операций сравнений; количество операций перестановок. Показать зависимость от мощности данных.

**3. Провести анализ прямых методов сортировок для статического списка (для тех же данных и параметров). Обоснованный вывод по результатам анализа.**

**4. Провести анализ прямых методов сортировок для односвязного линейного списка (для тех же данных и параметров). Обоснованный вывод по результатам анализа.**

**5. Провести анализ прямых методов сортировок для двусвязного линейного списка (для тех же данных и параметров).**

**Обоснованный вывод по результатам анализа.**

**6. Результаты анализа сортировок для структур данных (массив, статический список, связанные списков) представить в виде таблицы Excel и графиков**

**7. Провести сравнительный анализ прямых методов сортировок для всех выполненных структур данных. Обоснованный вывод по результатам анализа.**

**8. Выполнить исследование прямых методов сортировки (выбор, вставки, обмен, шейкер) для текстового файла. Обоснованный вывод по результатам анализа.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1000 | 10000 | 100000 |
|
| Выбором | Время | 0 | 0 | 0 | 0.003 | 0.174 | 15.24 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |
| Кол-во перестановок | 0 | 0 | 0 | 0 | 0 | 0 |
| Вставками | Время | 0 | 0 | 0 | 0 | 0.001 | 0.001 |
| Кол-во сравнений | 9 | 49 | 99 | 999 | 9999 | 99999 |
| Кол-во перестановок | 0 | 0 | 0 | 0 | 0 | 0 |
| Обменом | Время | 0 | 0 | 0 | 0.004 | 0.345 | 34.976 |
| Кол-во сравнений | 90 | 2450 | 9900 | 999000 | 99990000 | 1409965408 |
| Кол-во перестановок | 0 | 0 | 0 | 0 | 0 | 0 |
| Шейкерная | Время | 0 | 0 | 0 | 0.004 | 0.2257 | 18.916 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |
| Кол-во перестановок | 0 | 0 | 0 | 0 | 0 | 0 |

**Лучший случай для массива:**

**Анализ сортировок для лучшего случая.**

**Пузырьковая** сортировка является худшей из простых сортировок по времени, потому что этот метод сортировки имеет наибольшее количество сравнений для любого N (размер массива).

**Шейкерная** сортировка вторая по времени сразу после **пузырька** с разницей в 16 секунд из-за меньшего количества сравнений, в отличии от **пузырькового** метода, но медленнее методов **выбора** и **вставками**, из – за того что он ходит вперед в поисках максимального и обратно в поисках минимального элемента попутно переставляя элементы и уменьшая диапазон поиска и перестановок.

Метод **вставками** имеет наименьшее число сравнений для любого N, а также самый эффективный по времени, например, при методе вставками N = 100000 метод занимает 0,001 секунды, из – за того что он ходит вперед в определенном диапазоне и сортируя его, а так как у нас массив полностью упорядочен, он просто сравнивает их.

Метод **выбором** занимает 15,24 секунды для N = 100000, он ищет минимальный элемент и ставит его в первый индекс и уменьшает диапазон поиска меньшего элемента.

**Худший случай для массива:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1000 | 10000 | 100000 |
|
| Выбором | Время | 0 | 0 | 0 | 0.003 | 0.244 | 18.082 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |
| Кол-во перестановок | 5 | 25 | 50 | 500 | 5000 | 50000 |
| Вставками | Время | 0 | 0 | 0 | 0.003 | 0.251 | 31.783 |
| Кол-во сравнений | 54 | 1274 | 5049 | 500499 | 50004999 | 705082703 |
| Кол-во перестановок | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |
| Обменом | Время | 0 | 0 | 0.001 | 0.077 | 5.912 | 697.943 |
| Кол-во сравнений | 90 | 2450 | 9900 | 999000 | 99990000 | 1409965408 |
| Кол-во перестановок | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |
| Шейкерная | Время | 0 | 0 | 0.001 | 0.057 | 5.32 | 601.245 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |
| Кол-во перестановок | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |

**Анализ сортировок для худшего случая.**

Самая долгая сортировка является сортировка **пузырьком** в любом случае, он занимает больше времени чем остальные, например, **пузырьковая** сортировка занимает 697,943 секунды в случае N = 100000, когда **шейкер** занимает 601,245 секунды из – за того что он переставляет с начала максимальный элемент в самый конец, а потом самый минимальный из конца в начало делая **N=N-1** перестановок при каждом ходе вперед или назад.

За ними идет сортировка **вставками** занимающая31.738 секунд в случае N = 100000 и 704982704 перестановок из – за того что он при каждом повторе главного цикла должен возвращаться назад чтобы отсортировать новоприбывшие элементы в диапазон сортировки.

Самый эффективная сортировкой оказалась сортировка **выбором**, она заняла 18.082 секунды и N/2 перестановок так как метод находит минимальный и переставляет его в начало массива.

**Средний случай для массива:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1000 | 10000 | 100000 |
|
| Выбором | Время | 0 | 0 | 0 | 0,002 | 0,134 | 14,863 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |
| Кол-во перестановок | 7 | 45 | 97 | 990 | 9993 | 99981 |
| Вставками | Время | 0 | 0 | 0 | 0 | 0,076 | 29,019 |
| Кол-во сравнений | 34 | 720 | 2519 | 247553 | 25351217 | 2506144510 |
| Кол-во перестановок | 25 | 671 | 2420 | 246554 | 25341218 | 2506044511 |
| Обменом | Время | 0 | 0 | 0 | 0,03 | 3,067 | 343,571 |
| Кол-во сравнений | 90 | 2450 | 9900 | 999000 | 99990000 | 1409965408 |
| Кол-во перестановок | 25 | 671 | 2420 | 246554 | 25341218 | 2506044511 |
| Шейкерная | Время | 0 | 0 | 0 | 0,026 | 2,996 | 399,012 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 704982704 |
| Кол-во перестановок | 25 | 671 | 2420 | 246554 | 25341218 | 2506044511 |

**Анализ сортировок для среднего случая.**

Самый медленный метод – **пузырьковый** метод, занимающий 399 секунды, для N = 100000. За ним идет **шейкерная** сортировка которая имеет одинаковое количество сравнений и перестановок с **пузырьковой** и быстрее **пузырьковой** на 56 секунд.

После идет метод **вставками**, который занимает 29 секунды, хоть и количество перестановок одинаковые с **шейкерной** сортировкой.

Самый эффективный метод, это сортировка **выбором** занимающая 14.863 секунды для N = 100000, и имеющая меньшее количество перестановок.

**Лучший случай для односвязного списка:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1000 | 10000 | 100000 |
| Выбором | Время | 0 | 0 | 0,001 | 0,003 | 0,315 | 175,59 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 4999950000 |
| Кол-во перестановок | 0 | 0 | 0 | 0 | 0 | 0 |
| Вставками | Время | 0 | 0 | 0 | 0,003 | 0,334 | 200,654 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 4999950000 |
| Кол-во перестановок | 0 | 0 | 0 | 0 | 0 | 0 |
| Обмен | Время | 0 | 0 | 0 | 0,007 | 0,716 | 392,116 |
| Кол-во сравнений | 90 | 2450 | 9900 | 999000 | 99990000 | 9999900000 |
| Кол-во перестановок | 0 | 0 | 0 | 0 | 0 | 0 |

**Анализ сортировок для лучшего случая.**

**Пузырьковая** сортировка также, как и с массивом является худшей из простых сортировок по времени занимающая 392.116 секунд для N = 100000.

Метод **вставками** имеет после **пузырька** по затраченному времени, и занимает 200.654 секунда для N = 100000. Метод **выбора** является лучшим для проверки среди трех сортировок, и занимает 175.59 секунд для N = 100000 при таком же количестве сравнений как у метода **вставками**.

**Худший случай для односвязного списка:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1000 | 10000 | 100000 |
| Выбором | Время | 0 | 0 | 0 | 0,003 | 0,402 | 199,12 |
| Кол-во сравнений | 45 | 1225 | 4950 | 499500 | 49995000 | 4999950000 |
| Кол-во перестановок | 25 | 625 | 2500 | 250 000 | 25000000 | 2500000000 |
| Вставками | Время | 0 | 0 | 0 | 0 | 0 | 0,017 |
| Кол-во сравнений | 9 | 49 | 99 | 999 | 9999 | 99999 |
| Кол-во перестановок | 9 | 49 | 99 | 999 | 9999 | 99999 |
| Обмен | Время | 0 | 0 | 0,001 | 0,065 | 9,157 | 2318,9 |
| Кол-во сравнений | 72 | 2352 | 9702 | 997002 | 99970002 | 9999700002 |
| Кол-во перестановок | 36 | 1176 | 4851 | 498501 | 49985001 | 4999850001 |

**Анализ сортировок для худшего случая.**

Самая долгая сортировка является сортировка **пузырьком** в любом случае, он занимает больше времени чем остальные, например, **пузырьковая** сортировка занимает 2318,9 секунды в случае N = 100000.

За ним идет сортировка **выбором** занимающая199.12 секунд в случае N = 100000 и 2500000000 перестановок.

Самый эффективная сортировкой оказалась сортировка **вставками**, она заняла 0,017 секунды и N – 1 и N - 1 перестановок.

**Средний случай для односвязного списка:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Метод сортировки | N | 10 | 50 | 100 | 1000 | 10000 | 100000 |
| Выбором | Время | 0 | 0 | 0 | 0,002 | 0,441 | 602,476 |
| Кол-во сравнений | 36 | 1176 | 4851 | 498501 | 49985001 | 4999850001 |
| Кол-во перестановок | 12 | 131 | 332 | 5202 | 74031 | 944465 |
| Вставками | Время | 0 | 0 | 0 | 0,001 | 0,183 | 344,743 |
| Кол-во сравнений | 23 | 567 | 2526 | 253557 | 24657140 | 2493808900 |
| Кол-во перестановок | 5 | 42 | 95 | 990 | 9990 | 99990 |
| Обмен | Время | 0 | 0 | 0 | 0,029 | 3,913 | 870,406 |
| Кол-во сравнений | 64 | 2304 | 9604 | 996004 | 99960004 | 9999600004 |
| Кол-во перестановок | 18 | 639 | 2415 | 245886 | 25332592 | 2505988448 |

**Анализ сортировок для среднего случая.**

Самый медленный метод – **пузырьковый** метод, занимающий 870.406 секунды, для N = 100000. За ним идет сортировка, **выбором** которая быстрее **пузырьковой** на 268 секунд.

Самым эффективным методом оказалось сортировка **вставками**, который занимает 344.743 секунды и с количеством перестановок N – 10 для N = 100000.

**Анализ для пузырьковой сортировки**

В лучшем случае сложность алгоритма будет O(n). В среднем случае производительность также улучшится. Принимая во внимание, что начальный алгоритм был O(n2) для всех случаев. Bubble Sort действительно хорошо работает в лучшем случае, но в среднем и наихудшем случае, так как сравнение увеличивается, мы теряем время, и для завершения сортировки требуется очень много времени, даже если она оптимизирована.

**Анализ для сортировки выбором**

Алгоритм сортировки не подходит для больших наборов данных, поскольку его средняя и наихудшая случаи составляют Ο(n2). Подобные сравнения имеют почти одинаковую ситуацию во всех случаях, что не очень хорошо, так как это занимает много времени.

**Анализ для сортировки вставками**

Этот метод в большинстве случаев лучше двух предыдущих. Он адаптивный, то есть эффективный для наборов данных, которые уже существенно отсортированы. Время работы алгоритма сортировки вставками — это сумма времён работы каждого шага. Сложность по времени равна O(n2). Это заметно в лучшем случае, когда мы сравниваем его с квадратичными (Bubble, Selection) сортами. Но в среднем и худшем случае это работает почти одинаково.

**Анализ для сортировки шейкера**

Производительность в наихудшем случае - O(n2), производительность в лучшем случае - O (n), средняя производительность - O (n2), мы можем видеть, что это оптимизированная версия пузырьковой сортировки и он работает немного лучше, чем пузырьковая сортировка с массивами размером до 100000. После этого он показывает нам лучший результат сравнивая Bubble sort, но не лучший среди всех.