МИНОБРНАУКИ РОССИИ САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**ОТЧЕТ**

по лабораторной работе № 1

дисциплина «Алгоритмы и Структуры данных»

Тема: «Алгоритмы сортировки сравнением»

Студент гр. 3351 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Фабер К.А.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пестерев Д.О.

Санкт-Петербург

2024

**Цель лабораторной работы: реализация алгоритмов сортировки сравнением и исследование их временной сложности.**

Теоретическая часть.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Алгоритм | Асимптотическая временная сложность | | | Асимптотическая пространственная сложность | | |
|  | Лучший случай | Средний случай | Худший случай | Лучший случай | Средний случай | Худший случай |
| Сортировка выбором |  |  |  |  |  |  |
| Сортировка вставками |  |  |  |  |  |  |
| Сортировка пузырьком |  |  |  |  |  |  |
| Сортировка слиянием |  |  |  |  |  |  |
| Сортировка Шелла |  |  |  |  |  |  |
| Сортировка Шелла по Хиббарду |  |  |  |  |  |  |
| Сортировка Шелла по Пратту |  |  |  |  |  |  |
| Быстрая сортировка |  |  |  |  |  |  |
| Пирамидальная сортировка |  |  |  |  |  |  |

**Сортировка выбором.**

Суть алгоритма заключается в том, чтобы в каждой итерации цикла по элементам массива, находить минимальный элемент в неотсортированной части и производить обмен значениями с текущим элементом массива. Сортировка является неустойчивой.

Функция временной сложности и её асимптотическая оценка:

Худший случай:

Лучший случай:

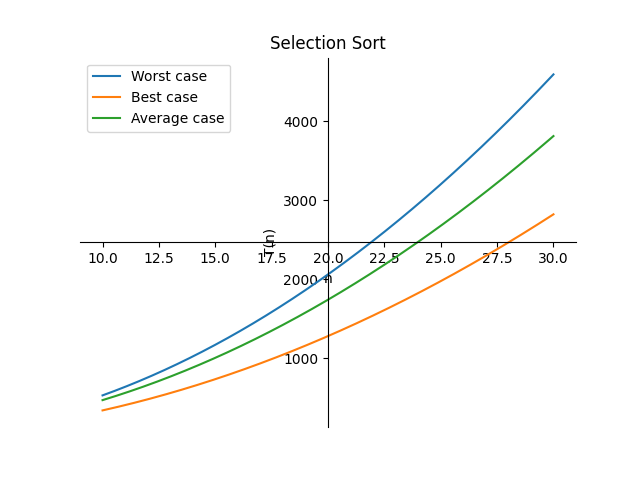
Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

Рисунок 1 – График функции временной сложности сортировки выбором

**Сортировка вставками.**

Алгоритм делит массив на отсортированную и неотсортированную части. Начиная с первого элемента, и по мере перебора элементов из неотсортированной части, они вставляются в правильное место в отсортированной части. Сортировка вставками является устойчивой.

Функция временной сложности и её асимптотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

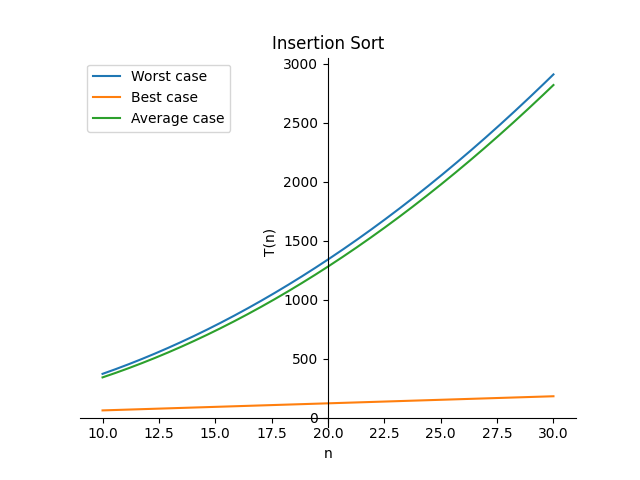
**

Рисунок 2 – График функции временной сложности сортировки вставками

**Сортировка пузырьком.**

Алгоритм проходит по массиву несколько раз, сравнивая соседние элементы и меняя их местами, если они расположены в неправильном порядке. Процесс повторяется, пока не будет сделан полный проход без перестановок, что означает, что массив отсортирован. Сортировка пузырьком является устойчивой.

Функция временной сложности и её асимптотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

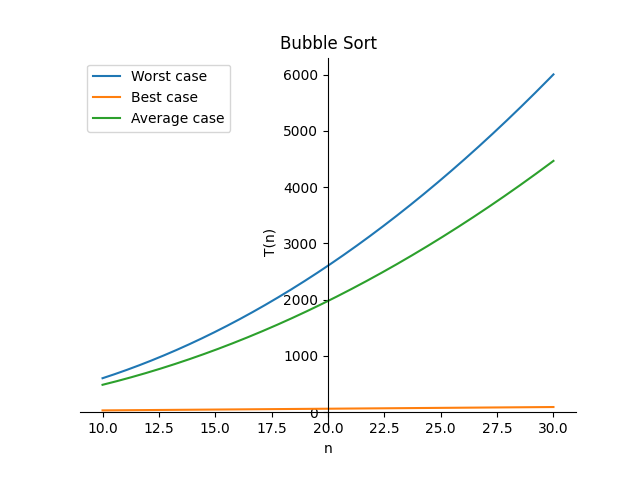


Рисунок 3 – График функции временной сложности сортировки пузырьком

**Сортировка слиянием.**

Алгоритм делит массив на две половины, рекурсивно сортирует каждую половину, а затем объединяет их обратно в отсортированный массив. Сортировка слиянием является устойчивой.

Функция временной сложности и её асимптотическая оценка:

Для нахождения функции временной сложности, используется мастер теорема:

Худший случай:

Найдем – функцию временной сложности слияния подмассивов:

Тогда,

Когда рекурсия прекращается, поэтому и на этом уровне , подставим k:

Лучший случай:

Аналогично:

Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Аналогично:

Средний случай:

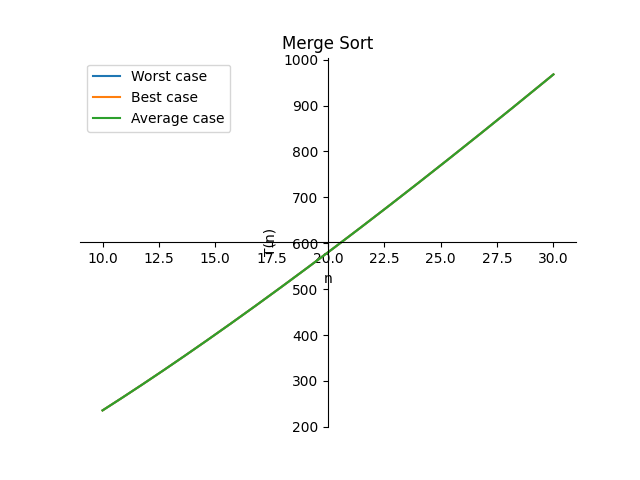
**

Рисунок 4 – График функции временной сложности сортировки слиянием

**Сортировка Шелла.**

Это обобщение сортировки вставками. Элементы сравниваются и сортируются на заданном расстоянии (или шаге), который с каждой итерацией уменьшается в 2 раза. Сортировка является неустойчивой

Функция временной сложности и её асимптотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

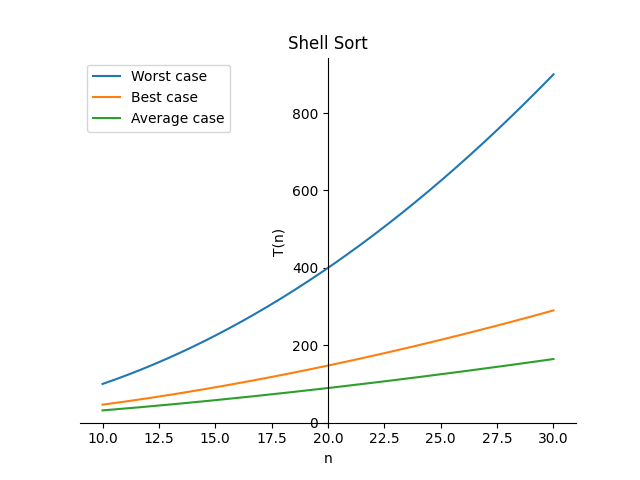


Рисунок 5 – График функции временной сложности сортировки Шелла

**Сортировка Шелла (последовательность Хиббарда).**

Вариант сортировки Шелла, где используется последовательность шагов . Шаги уменьшаются по этой последовательности. Сортировка является неустойчивой

Функция временной сложности и её асимптотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

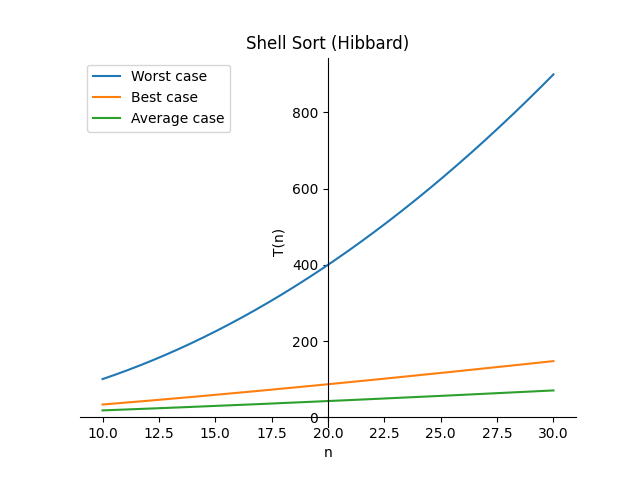


Рисунок 6 – График функции временной сложности сортировки Шелла (последовательность Хиббарда)

**Сортировка Шелла (последовательность Пратта).**

Вариант сортировки Шелла, где для шагов используется последовательность чисел, которая получается как комбинация степеней двойки и тройки:, где и — неотрицательные целые числа.

Функция временной сложности и её асимптотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

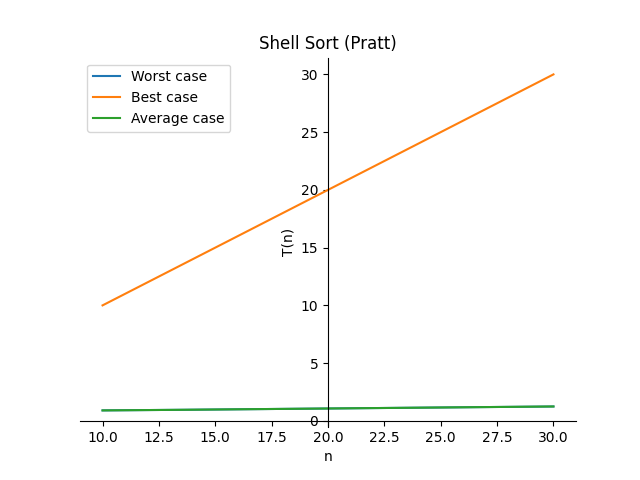


Рисунок 7 – График функции временной сложности сортировки Шелла(последовательность Пратта)

**Быстрая сортировка.**

Алгоритм выбирает опорный элемент и разделяет массив на элементы меньше и больше опорного, затем рекурсивно сортирует обе части.

Функция временной сложности и её асимптотическая оценка:

Худший случай:

Лучший случай:

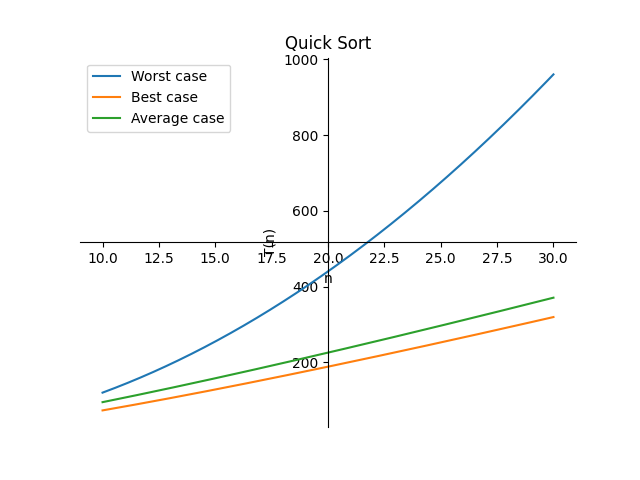
Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

Рисунок 8 – График функции временной сложности быстрой сортировки

**Пирамидальная сортировка.**

Сначала строится структура данных в виде пирамиды (кучи), затем на каждом шаге извлекается максимальный элемент и восстанавливается структура кучи. Сортировка является неустойчивой

Функция временной сложности и её асимптотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

Функция простраственной сложности и её асимтотическая оценка:

Худший случай:

Лучший случай:

Средний случай:

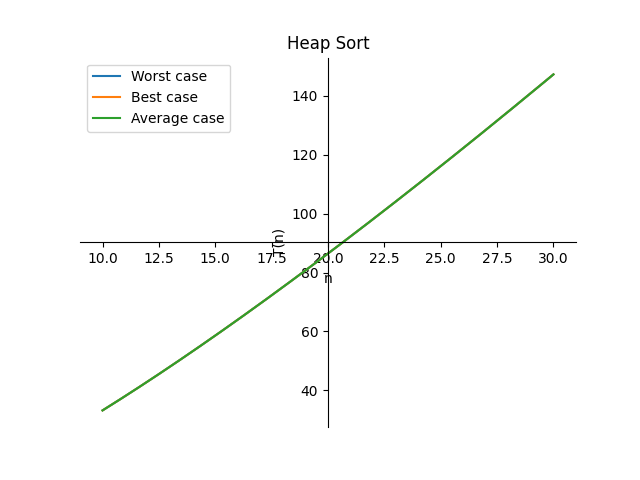
**

Рисунок 9 – График функции временной сложности пирамидальной сортировки

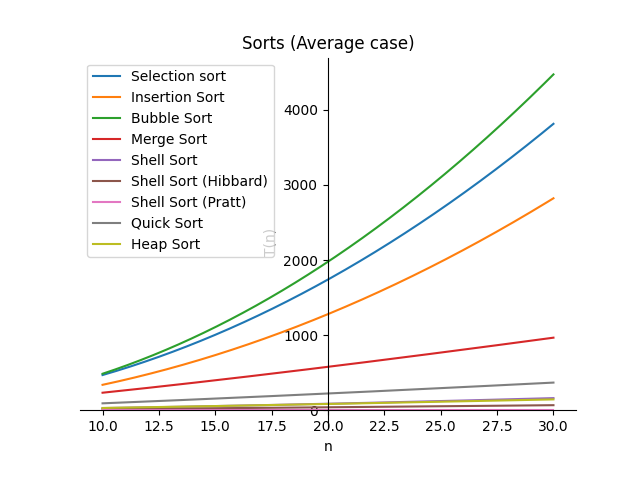


Рисунок 10 – Графики функции временной сложности всех сортировок в среднем случае

Исходя из теоретических данных, можно предположить что самой быстрой сортировой в среднем случае при размере массива > 100 000 будет сортировка Шелла с последовательностью Пратта.

**Практическая часть.**

Для выполнения практической части использовался язык C++, где были написаны все сортировки и функции заполнения, а также язык Python для отрисовки графиков. Для каждой сортировки и для каждого заполнения считалось время работы функции на определенном интервале элементов, результаты были представлены в таблице и на графиках.

**Результаты эксперимента:**

**Сортировка выбором.**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 2.66567 |
| 30000 | 4.66885 |
| 40000 | 6.39783 |
| 50000 | 8.94413 |
| 60000 | 11.4693 |
| 70000 | 15.6198 |
| 80000 | 20.381 |
| 90000 | 25.8344 |
| 100000 | 31.8396 |
| 110000 | 38.7964 |

Таблица 1 – Время работы сортировки выбором (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 1.2919 |
| 30000 | 2.91443 |
| 40000 | 5.21007 |
| 50000 | 8.15979 |
| 60000 | 11.7337 |
| 70000 | 15.9747 |
| 80000 | 20.8411 |
| 90000 | 26.414 |
| 100000 | 32.608 |
| 110000 | 39.4108 |

Таблица 2 – Время работы сортировки выбором (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 1.62244 |
| 30000 | 3.66446 |
| 40000 | 6.5082 |
| 50000 | 10.1608 |
| 60000 | 14.6623 |
| 70000 | 19.9365 |
| 80000 | 26.0528 |
| 90000 | 33.2684 |
| 100000 | 40.6719 |
| 110000 | 49.1756 |

Таблица 3 – Время работы сортировки выбором (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 1.26856 |
| 30000 | 2.87571 |
| 40000 | 5.11884 |
| 50000 | 7.98469 |
| 60000 | 11.4606 |
| 70000 | 15.633 |
| 80000 | 20.4218 |
| 90000 | 25.8532 |
| 100000 | 31.8553 |
| 110000 | 38.6172 |

Таблица 4 – Время работы сортировки выбором (случайный массив)

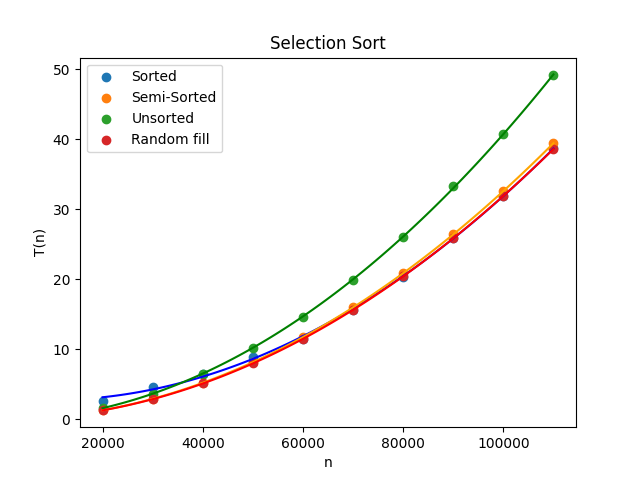
****

Рисунок 11 – График сортировки выбором для всех перечисленных случаев

**Сортировка вставками.**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0002023 |
| 30000 | 0.0002938 |
| 40000 | 0.0003852 |
| 50000 | 0.0005148 |
| 60000 | 0.0006166 |
| 70000 | 0.0006631 |
| 80000 | 0.0008135 |
| 90000 | 0.0008764 |
| 100000 | 0.0009646 |
| 110000 | 0.0010791 |

Таблица 5 – Время работы сортировки вставками (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.496464 |
| 30000 | 1.12121 |
| 40000 | 1.99586 |
| 50000 | 3.10445 |
| 60000 | 4.46802 |
| 70000 | 6.08987 |
| 80000 | 7.95875 |
| 90000 | 10.0981 |
| 100000 | 12.4482 |
| 110000 | 14.9828 |

Таблица 6 – Время работы сортировки вставками (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 2.77427 |
| 30000 | 6.21777 |
| 40000 | 11.0594 |
| 50000 | 17.3814 |
| 60000 | 24.8324 |
| 70000 | 33.8663 |
| 80000 | 44.2537 |
| 90000 | 55.8205 |
| 100000 | 68.9507 |
| 110000 | 83.4295 |
|  |  |

Таблица 7 – Время работы сортировки вставками (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 1.36942 |
| 30000 | 3.10041 |
| 40000 | 5.54315 |
| 50000 | 8.62329 |
| 60000 | 12.4247 |
| 70000 | 16.9918 |
| 80000 | 22.2321 |
| 90000 | 28.0488 |
| 100000 | 34.6441 |
| 110000 | 41.7512 |

Таблица 8 – Время работы сортировки вставками (случайный массив)

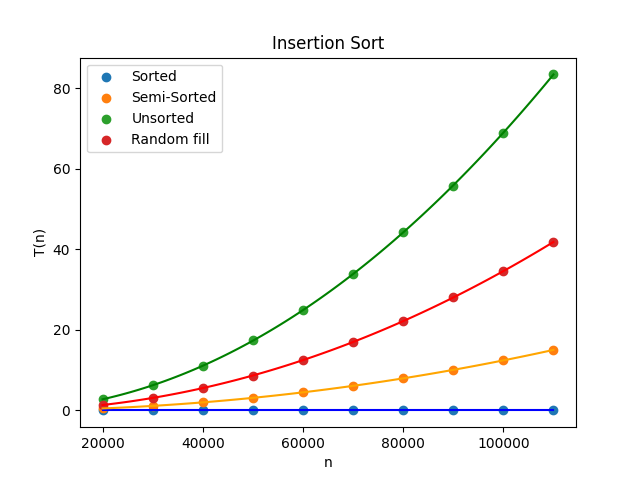


Рисунок 12 – График сортировки вставками для всех перечисленных случаев

**Сортировка пузырьком.**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0001824 |
| 30000 | 0.0002716 |
| 40000 | 0.0003632 |
| 50000 | 0.0004531 |
| 60000 | 0.0005587 |
| 70000 | 0.0006304 |
| 80000 | 0.0007265 |
| 90000 | 0.0008174 |
| 100000 | 0.0009131 |
| 110000 | 0.0010541 |

Таблица 9 – Время работы сортировки пузырьком (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 2.42518 |
| 30000 | 5.47569 |
| 40000 | 9.69952 |
| 50000 | 15.1398 |
| 60000 | 21.9849 |
| 70000 | 29.7326 |
| 80000 | 38.733 |
| 90000 | 49.2102 |
| 100000 | 60.6163 |
| 110000 | 73.242 |

Таблица 10 – Время работы сортировки пузырьком (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 4.73915 |
| 30000 | 10.6544 |
| 40000 | 18.9611 |
| 50000 | 29.8437 |
| 60000 | 42.8215 |
| 70000 | 58.3135 |
| 80000 | 76.1058 |
| 90000 | 96.4762 |
| 100000 | 118.597 |
| 110000 | 143.577 |

Таблица 11 – Время работы сортировки пузырьком (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 3.89553 |
| 30000 | 8.7496 |
| 40000 | 15.5626 |
| 50000 | 24.3244 |
| 60000 | 34.961 |
| 70000 | 47.7517 |
| 80000 | 62.5312 |
| 90000 | 78.6228 |
| 100000 | 97.0987 |
| 110000 | 117.903 |

Таблица 12 – Время работы сортировки пузырьком (случайный массив)

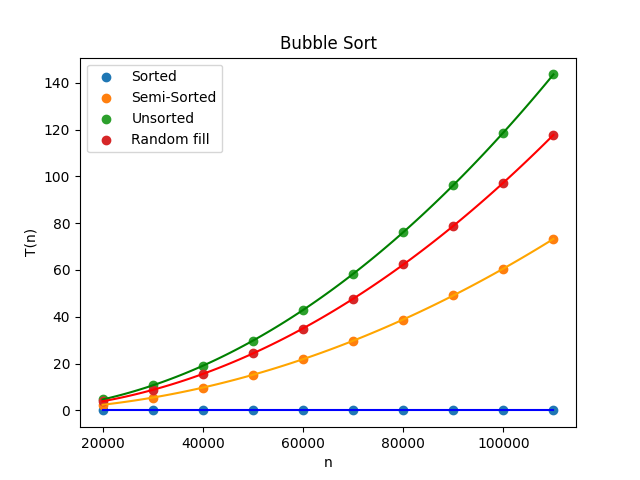


Рисунок 13 – График сортировки пузырьком для всех перечисленных случаев

**Сортировка слиянием.**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.119119 |
| 30000 | 0.181034 |
| 40000 | 0.241704 |
| 50000 | 0.302832 |
| 60000 | 0.367303 |
| 70000 | 0.419884 |
| 80000 | 0.486093 |
| 90000 | 0.544194 |
| 100000 | 0.609719 |
| 110000 | 0.670313 |

Таблица 13 – Время работы сортировки слиянием (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.119785 |
| 30000 | 0.184277 |
| 40000 | 0.238275 |
| 50000 | 0.303816 |
| 60000 | 0.362658 |
| 70000 | 0.423836 |
| 80000 | 0.488236 |
| 90000 | 0.549497 |
| 100000 | 0.610446 |
| 110000 | 0.664803 |

Таблица 14 – Время работы сортировки слиянием (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.11924 |
| 30000 | 0.181112 |
| 40000 | 0.239262 |
| 50000 | 0.300966 |
| 60000 | 0.36414 |
| 70000 | 0.423604 |
| 80000 | 0.491852 |
| 90000 | 0.552898 |
| 100000 | 0.611672 |
| 110000 | 0.679772 |

Таблица 15 – Время работы сортировки слиянием (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.120628 |
| 30000 | 0.184345 |
| 40000 | 0.243965 |
| 50000 | 0.30647 |
| 60000 | 0.369017 |
| 70000 | 0.430038 |
| 80000 | 0.500635 |
| 90000 | 0.565148 |
| 100000 | 0.625679 |
| 110000 | 0.686129 |

Таблица 16 – Время работы сортировки слиянием (случайный массив)

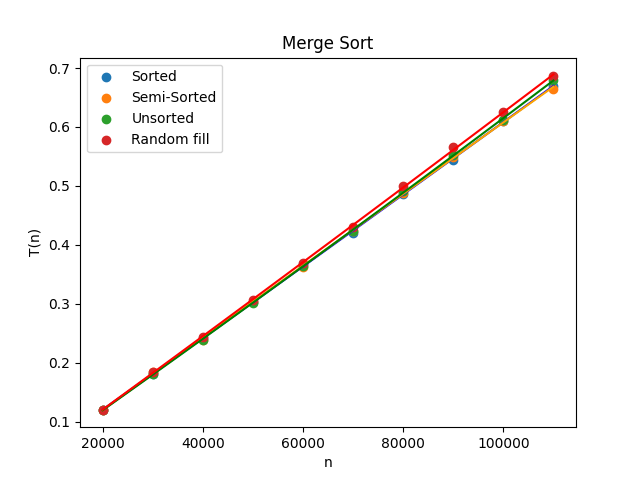


Рисунок 14 – График сортировки слиянием для всех перечисленных случаев

**Сортировка Шелла.**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0025692 |
| 30000 | 0.0040266 |
| 40000 | 0.0058938 |
| 50000 | 0.0070289 |
| 60000 | 0.0099503 |
| 70000 | 0.0109661 |
| 80000 | 0.0122907 |
| 90000 | 0.0141399 |
| 100000 | 0.0165586 |
| 110000 | 0.0173711 |

Таблица 17 – Время работы сортировки Шелла (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0046973 |
| 30000 | 0.0071715 |
| 40000 | 0.0098534 |
| 50000 | 0.0122118 |
| 60000 | 0.0159593 |
| 70000 | 0.0180952 |
| 80000 | 0.0214234 |
| 90000 | 0.0241003 |
| 100000 | 0.027351 |
| 110000 | 0.0295887 |

Таблица 18 – Время работы сортировки Шелла (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0062428 |
| 30000 | 0.0086927 |
| 40000 | 0.0124227 |
| 50000 | 0.0175604 |
| 60000 | 0.0198182 |
| 70000 | 0.0220982 |
| 80000 | 0.0264077 |
| 90000 | 0.0296526 |
| 100000 | 0.0344315 |
| 110000 | 0.0355962 |

Таблица 19 – Время работы сортировки Шелла (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0133668 |
| 30000 | 0.0196815 |
| 40000 | 0.0292328 |
| 50000 | 0.0364038 |
| 60000 | 0.0448961 |
| 70000 | 0.0542839 |
| 80000 | 0.0696328 |
| 90000 | 0.0741254 |
| 100000 | 0.0842963 |
| 110000 | 0.0930418 |

Таблица 20 – Время работы сортировки Шелла (случайный массив)

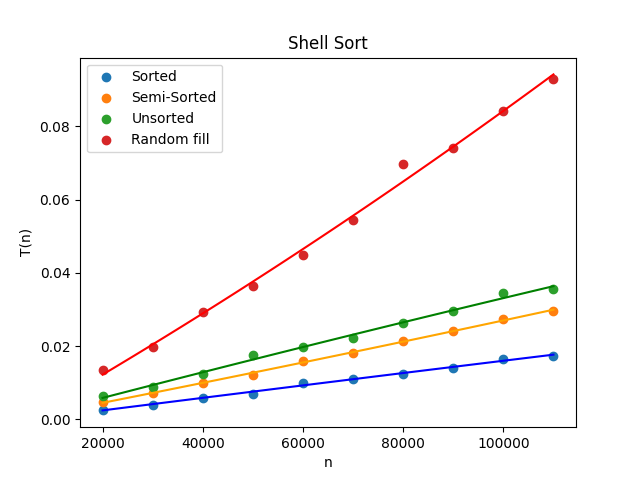


Рисунок 15 – График сортировки Шелла для всех перечисленных случаев

**Сортировка Шелла (последовательность Хиббарда).**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0026784 |
| 30000 | 0.004077 |
| 40000 | 0.0057306 |
| 50000 | 0.00707 |
| 60000 | 0.0097337 |
| 70000 | 0.0118632 |
| 80000 | 0.0125999 |
| 90000 | 0.0140554 |
| 100000 | 0.0153364 |
| 110000 | 0.0172229 |

Таблица 21 – Время работы сортировки Шелла с последовательностью Хиббарда (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.349606 |
| 30000 | 0.77396 |
| 40000 | 1.36914 |
| 50000 | 2.13966 |
| 60000 | 3.08466 |
| 70000 | 4.18427 |
| 80000 | 5.47437 |
| 90000 | 6.91487 |
| 100000 | 8.5398 |
| 110000 | 10.35 |

Таблица 22 – Время работы сортировки Шелла с последовательностью Хиббарда (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 1.3813 |
| 30000 | 3.08412 |
| 40000 | 5.48161 |
| 50000 | 8.51795 |
| 60000 | 12.2955 |
| 70000 | 16.794 |
| 80000 | 21.9399 |
| 90000 | 27.7016 |
| 100000 | 34.0814 |
| 110000 | 41.3485 |

Таблица 23 – Время работы сортировки Шелла с последовательностью Хиббарда (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.793968 |
| 30000 | 1.76007 |
| 40000 | 3.14114 |
| 50000 | 4.94465 |
| 60000 | 7.07201 |
| 70000 | 9.64302 |
| 80000 | 12.536 |
| 90000 | 15.8554 |
| 100000 | 19.5989 |
| 110000 | 23.8815 |

Таблица 24 – Время работы сортировки Шелла с последовательностью Хиббарда (случайный массив)

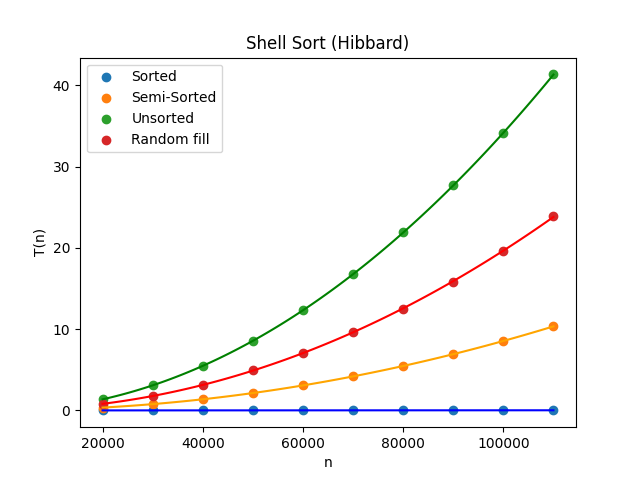


Рисунок 16 – График сортировки Шелла с последовательностью Хиббарда для всех перечисленных случаев

**Сортировка Шелла (последовательность Пратта).**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0147609 |
| 30000 | 0.0240634 |
| 40000 | 0.0338253 |
| 50000 | 0.0449055 |
| 60000 | 0.0549618 |
| 70000 | 0.0665212 |
| 80000 | 0.0777683 |
| 90000 | 0.0898924 |
| 100000 | 0.101281 |
| 110000 | 0.112626 |

Таблица 25 – Время работы сортировки Шелла с последовательностью Пратта (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.401001 |
| 30000 | 0.905097 |
| 40000 | 1.58703 |
| 50000 | 2.46637 |
| 60000 | 3.56336 |
| 70000 | 4.81731 |
| 80000 | 6.30777 |
| 90000 | 7.94743 |
| 100000 | 9.81106 |
| 110000 | 11.8715 |

Таблица 26 – Время работы сортировки Шелла с последовательностью Пратта (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 2.17904 |
| 30000 | 4.86932 |
| 40000 | 8.64304 |
| 50000 | 13.5119 |
| 60000 | 19.526 |
| 70000 | 26.5711 |
| 80000 | 34.6417 |
| 90000 | 43.9067 |
| 100000 | 55.3354 |
| 110000 | 65.4773 |

Таблица 27 – Время работы сортировки Шелла с последовательностью Пратта (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 1.10081 |
| 30000 | 2.46073 |
| 40000 | 4.35058 |
| 50000 | 6.76874 |
| 60000 | 9.75255 |
| 70000 | 13.3869 |
| 80000 | 17.4461 |
| 90000 | 22.0458 |
| 100000 | 27.2873 |
| 110000 | 32.8196 |

Таблица 28 – Время работы сортировки Шелла с последовательностью Пратта (случайный массив)

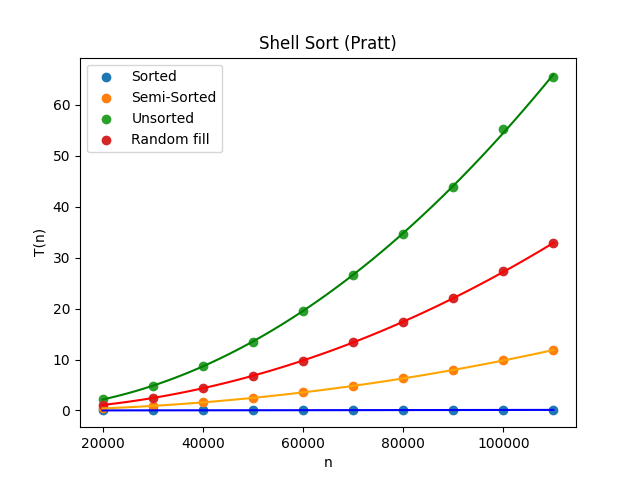


Рисунок 17 – График сортировки Шелла с последовательностью Пратта для всех перечисленных случаев

**Быстрая сортировка.**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 3.66786 |
| 30000 | 8.24835 |
| 40000 | 14.696 |
| 50000 | 22.916 |
| 60000 | 33.0002 |
| 70000 | 44.8831 |
| 80000 | 58.6188 |
| 90000 | 74.3403 |
| 100000 | 91.5042 |
| 110000 | 110.956 |

Таблица 29 – Время работы быстрой сортировки (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.285622 |
| 30000 | 0.68641 |
| 40000 | 1.13633 |
| 50000 | 1.94769 |
| 60000 | 2.72248 |
| 70000 | 3.59794 |
| 80000 | 4.53337 |
| 90000 | 6.35956 |
| 100000 | 7.77131 |
| 110000 | 9.38359 |

Таблица 30 – Время работы быстрой сортировки (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 2.23928 |
| 30000 | 5.02467 |
| 40000 | 8.96705 |
| 50000 | 13.982 |
| 60000 | 20.1267 |
| 70000 | 27.4946 |
| 80000 | 35.7306 |
| 90000 | 45.2961 |
| 100000 | 55.7968 |
| 110000 | 67.5421 |

Таблица 31 – Время работы быстрой сортировки (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0052804 |
| 30000 | 0.0078259 |
| 40000 | 0.0106528 |
| 50000 | 0.0143539 |
| 60000 | 0.0173637 |
| 70000 | 0.0223593 |
| 80000 | 0.0263657 |
| 90000 | 0.0267479 |
| 100000 | 0.0336292 |
| 110000 | 0.0348456 |

Таблица 32 – Время работы быстрой сортировки (случайный массив)

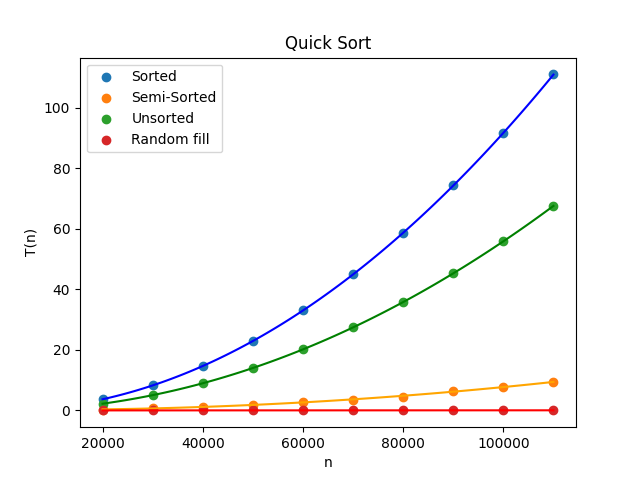


Рисунок 18 – График быстрой сортировки для всех перечисленных случаев

**Пирамидальная Сортировка.**

Отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0092998 |
| 30000 | 0.0155491 |
| 40000 | 0.0201777 |
| 50000 | 0.0273056 |
| 60000 | 0.0324229 |
| 70000 | 0.0377784 |
| 80000 | 0.0446009 |
| 90000 | 0.0513214 |
| 100000 | 0.0562167 |
| 110000 | 0.063118 |

Таблица 33 – Время работы пирамидальной сортировки (отсортированный массив)

Почти отсортированный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0091366 |
| 30000 | 0.0142432 |
| 40000 | 0.0201044 |
| 50000 | 0.0258544 |
| 60000 | 0.0328031 |
| 70000 | 0.0367901 |
| 80000 | 0.043337 |
| 90000 | 0.0494247 |
| 100000 | 0.0564179 |
| 110000 | 0.0606181 |

Таблица 34 – Время работы пирамидальной сортировки (почти отсортированный массив)

Отсортированный в сторону убывания массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0085953 |
| 30000 | 0.0138777 |
| 40000 | 0.0185811 |
| 50000 | 0.0252764 |
| 60000 | 0.0308708 |
| 70000 | 0.0351187 |
| 80000 | 0.0414403 |
| 90000 | 0.0462746 |
| 100000 | 0.0531185 |
| 110000 | 0.0595631 |

Таблица 35 – Время работы пирамидальной сортировки (массив отсортированный в сторону убывания)

Случайный массив

|  |  |
| --- | --- |
| Количество элементов | Время сортировки(сек) |
| 20000 | 0.0094293 |
| 30000 | 0.01591 |
| 40000 | 0.020359 |
| 50000 | 0.0261803 |
| 60000 | 0.032276 |
| 70000 | 0.0381665 |
| 80000 | 0.045444 |
| 90000 | 0.0523038 |
| 100000 | 0.0568849 |
| 110000 | 0.0643849 |

Таблица 36 – Время работы пирамидальной сортировки (случайный массив)

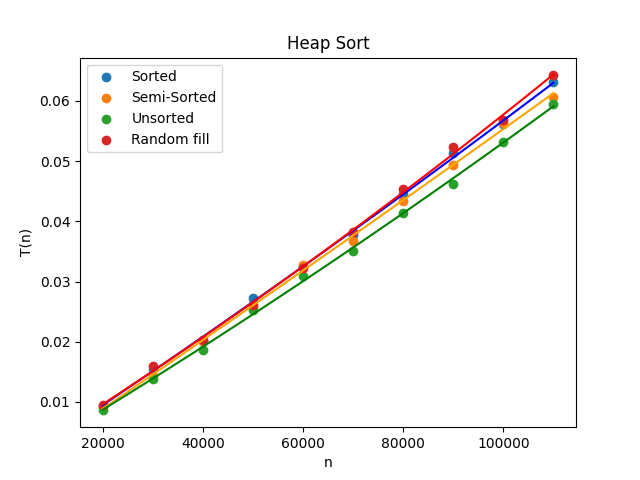


Рисунок 19 – График пирамидальной сортировки для всех перечисленных случаев

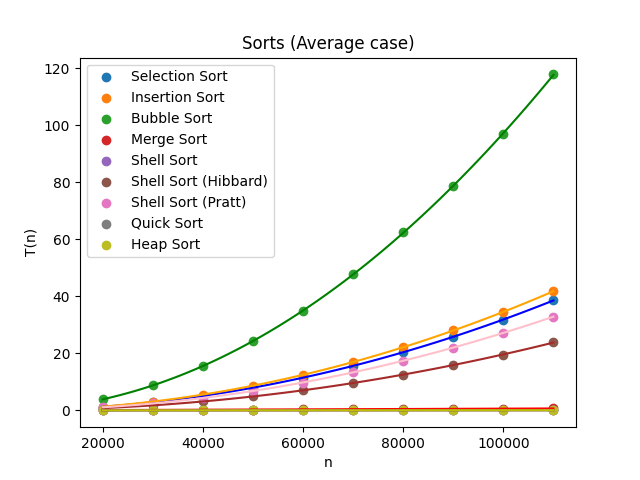


Рисунок 20 – График всех сортировок при случайном заполнении массива

**Вывод:**

Сравнивая теоретические и практические графики можно сделать вывод, что в большинстве случаев они согласовны и демонстрируют ожидаемую асимптотику, за исключением некоторых отклонений. Также, самой быстрой сортировкой в среднем случае при размере массива > 100 000 оказалась быстрая сортировка, что не совпадает с предположением из теоретической части.

**Ссылки:**

Репозиторий GitHub: