Санкт-Петербургский Государственный Технический Университет (Технологический институт)

Кафедра системного анализа и информационных технологий

**Лабораторная работа №4**

**Выполнил:**

**Андреев С.А**

**Проверил:**

**Мусаев А.А**

Санкт-Петербург, 2022

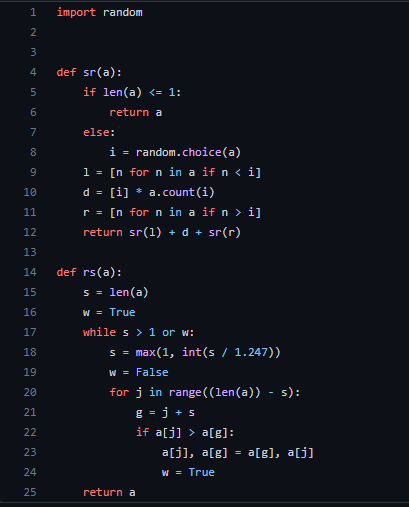
**Ход работы**

**Задание 1**

**“Написать программу для пузырьковой сортировки сложность данного метода оценить и сравнить с методом sort().”**

**Выполнение:**

**Я написал программу для сортировки методом “пузырька”.**



**В программе две функции:**

**1.SR – быстрая сортировка**

**2.RS – сортировка расческой**

**Далее я написал программу (код), которая использует данные функции как модуль. В этой программе пользователь выбирает один из двух предложенных методов сортировки и задает массив для дальнейшей сортировки выбранным методом. Программа, соответствуя введенным**

**данным, производит сортировку заданного массива одним из двух методов. Помимо этого,она оценивает время выполнения (скорость) с помощью timeit**

**Код программы представлен ниже на скриншоте и добавлен в репозиторий.**



**Задание 2**

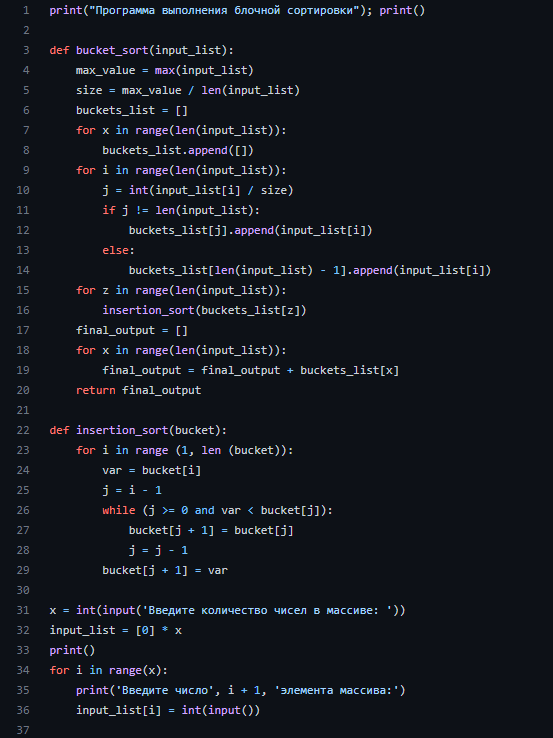
**“Изучить блочную и пирамидальную сортировку. Написать соответствующие программы”.**

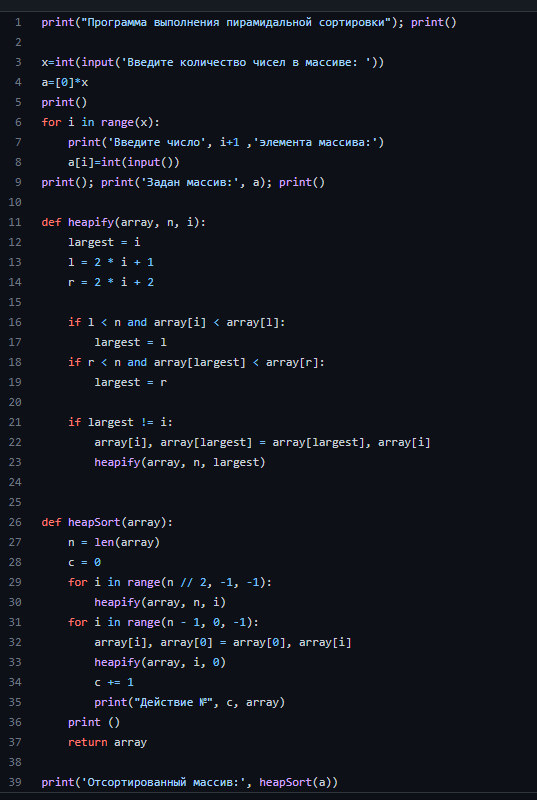
**Выполнение:**

**Пользуясь интернетом, я изучил методы блочной и**

**пирамидальной сортировок и выполнил задание Коды**

**программ представлены ниже и добавлены в репозиторий.**





**А)** Данная программа запрашивает у пользователя размерность массива, а затем значение каждого

элемента этого массива. На экран выводится заданный массива. Далее программа производит сортировку заданного массива блочным методом и выводит на экран уже отсортированный результат (массив)

Б) Данная программа запрашивает у пользователя размерность массива, а затем значение каждого элемента этого массива. На экран выводится заданный массив. Далее программа производит сортировку массива пирамидальным методом и выводит на экран каждую стадию сортировки. В конечном итоге на экран выводится отсортированный массив.

**Задание 3**

Оцените достоинства, недостатки и сложность изученных методов сортировок.”

Выполнение:

1. Быстрая сортировка:

a. Достоинства:

i. Один из самых быстродействующих (на практике) из алгоритмов

внутренней сортировки общего назначения.

ii. Требует лишь O(1) дополнительной памяти для своей работы

(неулучшенный рекурсивный алгоритм - в худшем случае O(n) памяти)

iii. Хорошо сочетается с механизмами кэширования и виртуальной памяти.

iv. Допускает естественное распараллеливание (сортировка выделенных

подмассивов в параллельно выполняющихся подпроцессах).

v. Допускает эффективную модификацию для сортировки по нескольким

ключам: благодаря тому, что в процессе разделения автоматически

выделяется отрезок элементов, равных опорному, этот отрезок можно

сразу же сортировать по следующему ключу.

vi. Работает на связных списках и других структурах с последовательным

доступом, допускающих эффективный проход как от начала к концу, так

и от конца к началу.

b. Недостатки:

i. Сильно деградирует по скорости (до O(n^2)) в худшем или близком к

нему случае, что может случиться при неудачных входных данных.

ii. Прямая реализация в виде функции с двумя рекурсивными вызовами

может привести к ошибке переполнения стека, так как в худшем случае

ей может потребоваться сделать O(n) вложенных рекурсивных вызовов.

iii. Неустойчив.

c. Сложность по времени:

i. Лучшее время: O(n log2 n)

ii. Среднее время: O(n log n)

iii. Худшее время: O(n^2)

d. Сложность по памяти

i. O(n)

ii. O(log n)

2. Сортировка расчёской:

a. Достоинства:

i. Достаточно простой алгоритм, построенный на основе пузырьковой

сортировки

b. Недостатки:

i. Скорость выполнения программы. Выполняется большое количество

перестановок, на что тратится много процессорного времени.

ii. Производит полный перебор всех элементов массива.

c. Сложность по времени:

i. Лучшее время: O(n log n)

ii. Среднее время: O(n^2 / 2^P)

iii. Худшее время: O(n^2)

d. Сложность по памяти:

i. O(n)

ii. O(1)

3. Блочная сортировка:

a. Достоинства:

i. Относится к классу быстрых алгоритмов с линейным временем

исполнения O(N) (на удачных входных данных).

b. Недостатки:

i. Сильно деградирует при большом количестве мало отличных

элементов, или же на неудачной функции получения номера корзины по

содержимому элемента.

c. Сложность по времени:

i. O(n)

Пирамидальная сортировка:

a. Достоинства:

i. Имеет доказанную оценку худшего случая O(n log n)

ii. Сортирует на месте, то есть требует всего O(1) дополнительной памяти.

b. Недостатки:

i. Неустойчив — для обеспечения устойчивости нужно расширять ключ.

ii. На почти отсортированных массивах работает столь же долго, как и на

хаотических данных.

iii. На одном шаге выборку приходится делать хаотично по всей длине

массива — поэтому алгоритм плохо сочетается с кэшированием и

подкачкой памяти.

iv. Методу требуется «мгновенный» прямой доступ; не работает на

связанных списках и других структурах памяти последовательного

доступа.

v. Не распараллеливается.

c. Сложность по времени:

i. O(n log n)

d. Сложность по памяти:

i. O(1)