Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Российский химико-технологический университет имени Д.И. Менделеева»
Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

Выполнил студент группы .....КС-38..... Прилепский Артем Сергеевич
Ссылка на репозиторий: .....github.com/news1d/Algorithms_and_structures

Приняли: ..... Пысин Максим Дмитриевич
.....Краснов Дмитрий Олегович
.....Лобанов Алексей Владимирович
.....Крашенинников Роман Сергеевич

Дата сдачи: 17.02.2023

Оглавление

Описание задачи.....3

Описание метода/модели.....3

Выполнение задачи.....3

Заключение.....5

Описание задачи.

Требуется реализовать алгоритм сортировки перемешиванием. Провести ряд замеров на массиве заполненном случайными дробными числами от -1.0 до 1.0. Размер входных данных для метода сортировки - 1000, 2000, 4000, 8000, 16000, 32000, 64000, 128000. Для каждого N сделать 20 замеров времени выполнения алгоритма. Полученные данные вывести на графике, сделать графики лучших попыток\худших\средних.

Описание метода/модели.

Сортировка перемешиванием (англ. *cocktail sort*), также известная как **Шейкерная сортировка** — разновидность пузырьковой сортировки, сортирующая массив в двух направлениях на каждой итерации. В среднем, сортировка перемешиванием работает в два раза быстрее пузырька. Сложность — $O(n^2)$, но стремится она к $O(k \cdot n)$, где k — максимальное расстояние элемента в неотсортированном массиве от его позиции в отсортированном массиве

Выполнение задачи.

Для реализации программы был выбран язык программирования Python. Программу можно условно разделить на 3 блока: блок заполнения массива случайными числами, сортировка массива, запись результатов работы в файл.

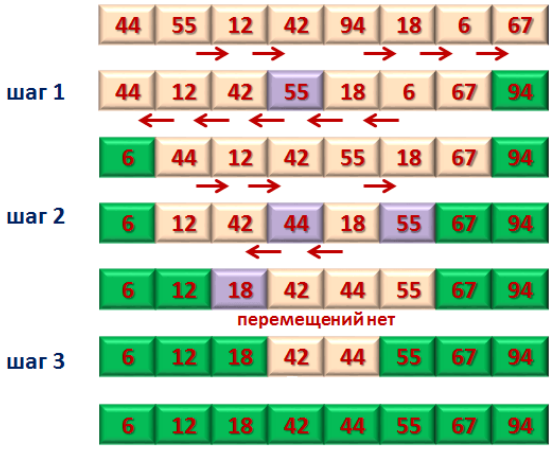
1. Заполнение массива случайными числами

   Для генерации случайных чисел мы используем модуль random в совокупности с методом uniform(), который позволяет задать нужный диапазон для генерации. Методом append() добавляем сгенерированное число в массив.

   ```
   for j in range(0, length):
       element = random.uniform(-1.0, 1.0)
       arr.append(element)
   ```

2. Сортировка

   Здесь мы проходимся сначала по массиву слева направо и сравниваем i элемент с i + 1. Если i > i + 1, то меняем их местами. Дойдя до конца массива у нас получается, что на N позицию встаёт самый максимальный элемент. Теперь мы двигаемся в направлении справа налево и сравниваем i элемент с i + 1. Если i > i + 1, то меняем их местами. Когда мы вернёмся в начало массива, наш минимальный элемент займёт нулевую позицию.




```
def shaker_sort(array):
    length = len(array)
    swapped = True
    start_index = 0
    end_index = length - 1

    while (swapped == True):
        swapped = False

        # проход слева направо
        for i in range(start_index, end_index):
            if (array[i] > array[i + 1]):
                # обмен элементов
                array[i], array[i + 1] = array[i + 1], array[i]
                swapped = True

        # если не было обменов прерываем цикл
        if (not (swapped)):
            break

        swapped = False
        end_index = end_index - 1

        # проход справа налево
        for i in range(end_index - 1, start_index - 1, -1):
            if (array[i] > array[i + 1]):
                # обмен элементов
                array[i], array[i + 1] = array[i + 1], array[i]
                swapped = True

        start_index = start_index + 1
```

3. Замер времени сортировки каждого массива мы производим с помощью класса timer(). Выводим результат работы в консоль и записываем в файл.

```
start = timer()
shaker_sort(arr)
end = timer()

my_file.write(f"{i + 1}. Sorting time: {round(end - start, 5)} seconds\n")
my_file.close()
```

Заключение.

В ходе лабораторной работы были сделаны замеры скорости работы алгоритма сортировки перемешиванием на наборах данных различного размера. Алгоритм имеет квадратичную сложность, что не является эффективным, т.к. на больших наборах данных сильно увеличивается время его работы.