САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №2

по курсу «Алгоритмы и структуры данных»

Тема: Сортировка слиянием. Метод декомпозиции
Вариант 1

Выполнил:
Бен Шамех Абделазиз
Группа: K3239

Проверила:
Ромакина Оксана Михайловна

Санкт-Петербург
2025 г.

Содержание

1 Задача 1. Сортировка слиянием

1.1 Условие

Используя псевдокод процедур Merge и Merge-sort, напишите программу сортировки слиянием на Python.

- **Вход:** Число n, массив из n целых чисел.
- **Выход:** Отсортированный массив.

1.2 Листинг кода

```python
def merge_sort(arr, p, r):
    if p < r:
        q = (p+r)//2
        merge_sort(arr,p,q)
        merge_sort(arr,q+1,r)
        merge(arr, p, q, r)

def merge(arr, p, q, r):
    arr1 = arr[p:q+1]
    arr2 = arr[q+1:r+1]
    i = j = 0
    while i < len(arr1) and j < len(arr2):
        if arr1[i] < arr2[j]:
            arr[p+i+j] = arr1[i]
            i += 1
        else:
            arr[p+i+j] = arr2[j]
            j += 1
    while i < len(arr1):
        arr[p+i+j] = arr1[i]
        i += 1
    while j < len(arr2):
        arr[p+i+j] = arr2[j]
        j += 1
```

1.3 Результаты выполнения

Пример из задачи:

- **Input (input.txt):** `6 \n 31 41 59 26 41 58`
- **Output (output.txt):** `26 31 41 41 58 59`

Метрики эффективности:

Сценарий	Время (сек)	Память (Мб)
Нижняя граница (n=1)	0.00022	15.54
Пример из задачи	0.00016	15.54
Верхняя граница	0.02904	16.51

2 Задача 2. Сортировка слиянием+

2.1 Условие

Выводить индексы граничных элементов и их значения после каждого слияния.

2.2 Результаты выполнения

Ввод: `10 \n 1 8 2 1 4 7 3 2 3 6`
Вывод:

```
1 2 1 8
4 5 1 4
4 6 1 7
... (промежуточные логи) ...
1 10 1 8
1 1 2 2 3 3 4 6 7 8
```

3 Задача 3. Число инверсий

3.1 Объяснение

Во время процесса слияния добавим к счетчику количество оставшихся элементов в левом массиве при перемещении элемента из правого массива.

3.2 Результаты выполнения

Пример:

- **Input:** `5 \n 2 3 9 2 9`
- **Output:** `2`

Метрики эффективности:

Сценарий	**Время (сек)**	**Память (Мб)**
Нижняя граница	0.0001	15.55
Пример из задачи	0.00015	15.54
Верхняя граница	0.18916	16.18

4 Задача 4. Бинарный поиск

4.1 Объяснение

Делим массив пополам и сравниваем центральное значение с ключом. Сложность $O(\log n)$.

4.2 Результаты выполнения

Метрики эффективности:

Сценарий	**Время (сек)**	**Память (Мб)**
Нижняя граница	0.00015	15.29
Пример из задачи	0.00025	15.32
Верхняя граница	0.00028	17.35

5 Задача 5. Представитель большинства

5.1 Условие

Проверить, содержит ли последовательность элемент, который появляется чаще, чем $n/2$ раз.

5.2 Результаты выполнения

Метрики эффективности:

Сценарий	**Время (сек)**	**Память (Мб)**
Нижняя граница	0.00022	15.40
Пример из задачи	0.00014	15.54
Верхняя граница	0.03461	16.92

6 Задача 6. Поиск максимальной прибыли

6.1 Результаты выполнения

Анализ акций (Пример Газпром):

Сценарий	**Время (сек)**	**Память (Мб)**
Нижняя граница	0.00017	15.62
Пример из задачи	0.00014	15.55
Верхняя граница	0.11067	23.22

7 Задача 7. Поиск максимального подмассива (Линейно)

7.1 Объяснение

Нахождение максимального подмассива за $O(n)$ путем отслеживания текущей суммы и минимального префикса.

7.2 Результаты выполнения

Метрики эффективности:

Сценарий	**Время (сек)**	**Память (Мб)**
Нижняя граница	0.00808	29.70
Пример из задачи	0.00916	29.62
Верхняя граница	0.10914	31.95

8 Вывод

В ходе выполнения лабораторной работы были достигнуты следующие результаты:

- Реализован и протестирован алгоритм сортировки слиянием (*Merge Sort*) со сложностью $O(n \log n)$.
- Изучен метод «Разделяй и властвуй», примененный для задач поиска инверсий и поиска представителя большинства.
- Реализован бинарный поиск, обеспечивающий логарифмическое время поиска.
- Проведено сравнение рекурсивного подхода и линейного алгоритма для поиска максимального подмассива, подтвердившее преимущество линейного времени $O(n)$.