


Лабораторная работа №1

Задание:



Российский университет
дружбы народов
RUDN University

ЗАДАНИЕ

Реализовать на любом из языков программирования алгоритм быстрой сортировки массива.

Программная реализация должна:

- содержать подробные комментарии по всем выполняемым действиям
- обеспечивать ввод массива произвольной длины с клавиатуры,
- выполнять подсчёт произведённых перестановок элементов
- отображать процесс сортировки (выводить на экран промежуточные состояния сортируемого массива после каждой произведённой перестановки).

Отчёт о выполнении лабораторной работы должен содержать:

- Описание основных компонентов программы
- Листинг программы в текстовом виде (в виде приложения к отчёту)
- Результаты работы программы на обязательном тестовом (4 9 1 5 4 8 3 7 9 4) и ещё двух произвольных наборах данных с указанием исходного и отсортированного массивов, выбираемых опорных элементов и количества произведённых в ходе сортировки перестановок со скриншотами.

Виноградов Андрей Николаевич vinogradov-an@rudn.ru ФМиЕН, Кафедра информационных технологий © 2020

13

Выполнение:

1. Подготовка

In [104... `!pwd`

`/Users/artym.petrov/dev/university/4-1/it-computer-practice/lab01`

In [105... `from typing import List`

2. Функция для быстрой сортировки

```
In [161... def quick_sort(arr: List[int], left: int, right: int, swap_count: int) -> int:
    if left >= right:
        return swap_count

    pivot = arr[right]

    if verbose:
        print(f"\nОпорный элемент: {pivot} (индекс {right})")
        print(f"Текущий подмассив: {arr[left:right+1]}")

    # Разделение
    i = left
    for j in range(left, right):
        if arr[j] < pivot:
```

```

        if i != j:
            arr[i], arr[j] = arr[j], arr[i]
            swap_count += 1
            if verbose:
                print(f"  Перестановка #{swap_count}: {arr}")
        i += 1

    # Ставим опорный на место
    if i != right:
        arr[i], arr[right] = arr[right], arr[i]
        swap_count += 1
        if verbose:
            print(f"  Перестановка #{swap_count}: {arr}")

    swap_count = quick_sort(arr, left, i - 1, swap_count, verbose)
    swap_count = quick_sort(arr, i + 1, right, swap_count, verbose)

    return swap_count

```

3. Функция для вывода массива

```

In [162... def print_array(arr: List[int], label: str = ""):
    if label:
        print(f"{label}: ", end="")
    print("[", end="")
    for i, val in enumerate(arr):
        if i > 0:
            print(", ", end="")
        print(val, end="")
    print("]")

```

4. Функция для чтения массива из stdin

```

In [165... def get_user_input() -> List[int]:
    print("Введите элементы массива через пробел:")
    try:
        user_input = input()
        arr = [int(x) for x in user_input.split()]
        return arr
    except ValueError:
        print("Ошибка! Вводите только целые числа, разделенные пробелом")
        return None

```

5. Запуск

```

In [166... test_arrays = [
    [4, 9, 1, 5, 4, 8, 3, 7, 9, 4],
    [7, 2, 1, 6, 8, 5, 3, 4],
    [10, 80, 30, 90, 40, 50, 70],
]

arrays_to_sort = []

```

```
print("\nВыберите способ ввода данных:")
print("1 – Использовать тестовые наборы (обязательный + 2 произвольных)")
print("2 – Ввести массив вручную")

choice = input("Ваш выбор (1 или 2): ")

if choice == "1":
    arrays_to_sort = test_arrays
    print("\nБудут использованы тестовые наборы данных:")
    for i, arr in enumerate(test_arrays):
        print(f"Набор {i+1}: {arr}")
elif choice == "2":
    arr = get_user_input()
    if arr is not None:
        arrays_to_sort = [arr]
    else:
        exit(0)
else:
    print("Неверный выбор. Используются тестовые наборы.")
    arrays_to_sort = test_arrays

for idx, original_arr in enumerate(arrays_to_sort):
    print("\n" + "=" * 60)
    print(f"СОРТИРОВКА НАБОРА {idx+1}")
    print("=" * 60)

    arr_to_sort = original_arr.copy()

    print_array(original_arr, "Исходный массив")
    print()

    swap_count = 0

    print("НАЧАЛО ПРОЦЕССА СОРТИРОВКИ:")
    swap_count = quick_sort(arr_to_sort, 0, len(arr_to_sort) - 1, s

    print("\n" + "-" * 40)
    print_array(original_arr, "Исходный массив")
    print_array(arr_to_sort, "Отсортированный массив")
    print(f"Количество перестановок: {swap_count}")
    print("-" * 40)

    if arr_to_sort == sorted(original_arr):
        print("Массив отсортирован правильно")
    else:
        print("Ошибка сортировки!")

    if idx < len(arrays_to_sort) - 1:
        input("\nНажмите Enter для продолжения...")
```

Выберите способ ввода данных:

- 1 – Использовать тестовые наборы (обязательный + 2 произвольных)
- 2 – Ввести массив вручную

Неверный выбор. Используются тестовые наборы.

=====

СОРТИРОВКА НАБОРА 1

=====

Исходный массив: [4, 9, 1, 5, 4, 8, 3, 7, 9, 4]

НАЧАЛО ПРОЦЕССА СОРТИРОВКИ:

Опорный элемент: 4 (индекс 9)

Текущий подмассив: [4, 9, 1, 5, 4, 8, 3, 7, 9, 4]

Перестановка #1: [1, 9, 4, 5, 4, 8, 3, 7, 9, 4]

Перестановка #2: [1, 3, 4, 5, 4, 8, 9, 7, 9, 4]

Перестановка #3: [1, 3, 4, 5, 4, 8, 9, 7, 9, 4]

Опорный элемент: 3 (индекс 1)

Текущий подмассив: [1, 3]

Опорный элемент: 4 (индекс 9)

Текущий подмассив: [5, 4, 8, 9, 7, 9, 4]

Перестановка #4: [1, 3, 4, 4, 4, 8, 9, 7, 9, 5]

Опорный элемент: 5 (индекс 9)

Текущий подмассив: [4, 8, 9, 7, 9, 5]

Перестановка #5: [1, 3, 4, 4, 4, 5, 9, 7, 9, 8]

Опорный элемент: 8 (индекс 9)

Текущий подмассив: [9, 7, 9, 8]

Перестановка #6: [1, 3, 4, 4, 4, 5, 7, 9, 9, 8]

Перестановка #7: [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Опорный элемент: 9 (индекс 9)

Текущий подмассив: [9, 9]

Перестановка #8: [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Исходный массив: [4, 9, 1, 5, 4, 8, 3, 7, 9, 4]

Отсортированный массив: [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Количество перестановок: 8

Массив отсортирован правильно

```
=====
СОРТИРОВКА НАБОРА 2
=====
```

Исходный массив: [7, 2, 1, 6, 8, 5, 3, 4]

НАЧАЛО ПРОЦЕССА СОРТИРОВКИ:

Опорный элемент: 4 (индекс 7)

Текущий подмассив: [7, 2, 1, 6, 8, 5, 3, 4]

Перестановка #1: [2, 7, 1, 6, 8, 5, 3, 4]

Перестановка #2: [2, 1, 7, 6, 8, 5, 3, 4]

Перестановка #3: [2, 1, 3, 6, 8, 5, 7, 4]

Перестановка #4: [2, 1, 3, 4, 8, 5, 7, 6]

Опорный элемент: 3 (индекс 2)

Текущий подмассив: [2, 1, 3]

Опорный элемент: 1 (индекс 1)

Текущий подмассив: [2, 1]

Перестановка #5: [1, 2, 3, 4, 8, 5, 7, 6]

Опорный элемент: 6 (индекс 7)

Текущий подмассив: [8, 5, 7, 6]

Перестановка #6: [1, 2, 3, 4, 5, 8, 7, 6]

Перестановка #7: [1, 2, 3, 4, 5, 6, 7, 8]

Опорный элемент: 8 (индекс 7)

Текущий подмассив: [7, 8]

```
-----
Исходный массив: [7, 2, 1, 6, 8, 5, 3, 4]
```

Отсортированный массив: [1, 2, 3, 4, 5, 6, 7, 8]

Количество перестановок: 7

```
-----
```

Массив отсортирован правильно

```
=====
СОРТИРОВКА НАБОРА 3
=====
```

Исходный массив: [10, 80, 30, 90, 40, 50, 70]

НАЧАЛО ПРОЦЕССА СОРТИРОВКИ:

Опорный элемент: 70 (индекс 6)

Текущий подмассив: [10, 80, 30, 90, 40, 50, 70]

Перестановка #1: [10, 30, 80, 90, 40, 50, 70]

Перестановка #2: [10, 30, 40, 90, 80, 50, 70]

Перестановка #3: [10, 30, 40, 50, 80, 90, 70]

Перестановка #4: [10, 30, 40, 50, 70, 90, 80]

Опорный элемент: 50 (индекс 3)

Текущий подмассив: [10, 30, 40, 50]

Опорный элемент: 40 (индекс 2)

Текущий подмассив: [10, 30, 40]

Опорный элемент: 30 (индекс 1)

Текущий подмассив: [10, 30]

Опорный элемент: 80 (индекс 6)

Текущий подмассив: [90, 80]

Перестановка #5: [10, 30, 40, 50, 70, 80, 90]

```
-----
```

Исходный массив: [10, 80, 30, 90, 40, 50, 70]

Отсортированный массив: [10, 30, 40, 50, 70, 80, 90]

Количество перестановок: 5

```
-----
```

Массив отсортирован правильно

Результат

1. Начальный экран

Выберите способ ввода данных:

1 – Использовать тестовые наборы (обязательный + 2 произвольных)

2 – Ввести массив вручную

Ваш выбор (1 или 2):

Неверный выбор. Используются тестовые наборы.

2. Результат сортировки первого массива

СОРТИРОВКА НАБОРА 1

Исходный массив: [4, 9, 1, 5, 4, 8, 3, 7, 9, 4]

НАЧАЛО ПРОЦЕССА СОРТИРОВКИ:

Опорный элемент: 4 (индекс 4)

Текущий подмассив: [4, 9, 1, 5, 4, 8, 3, 7, 9, 4]

Перестановка: [4, 9, 1, 5, 4, 8, 3, 7, 9, 4]

Перестановка: [3, 9, 1, 5, 4, 8, 4, 7, 9, 4]

Перестановка: [3, 4, 1, 5, 9, 8, 4, 7, 9, 4]

Перестановка (возврат опорного): [3, 4, 1, 4, 9, 8, 4, 7, 9, 5]

Позиция опорного элемента после разделения: индекс 3

Перестановка для выбора медианы: [1, 4, 3, 4, 9, 8, 4, 7, 9, 5]

Перестановка для выбора медианы: [1, 3, 4, 4, 9, 8, 4, 7, 9, 5]

Опорный элемент: 3 (индекс 1)

Текущий подмассив: [1, 3, 4]

Перестановка: [1, 4, 3, 4, 9, 8, 4, 7, 9, 5]

Перестановка (возврат опорного): [1, 3, 4, 4, 9, 8, 4, 7, 9, 5]

Позиция опорного элемента после разделения: индекс 1

Перестановка для выбора медианы: [1, 3, 4, 4, 4, 8, 9, 7, 9, 5]

Перестановка для выбора медианы: [1, 3, 4, 4, 4, 8, 5, 7, 9, 9]

Опорный элемент: 5 (индекс 6)

Текущий подмассив: [4, 8, 5, 7, 9, 9]

Перестановка: [1, 3, 4, 4, 4, 8, 9, 7, 9, 5]

Перестановка (возврат опорного): [1, 3, 4, 4, 4, 5, 9, 7, 9, 8]

Позиция опорного элемента после разделения: индекс 5

Перестановка для выбора медианы: [1, 3, 4, 4, 4, 5, 7, 9, 9, 8]

Перестановка для выбора медианы: [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Опорный элемент: 8 (индекс 7)

Текущий подмассив: [7, 8, 9, 9]

Перестановка: [1, 3, 4, 4, 4, 5, 7, 9, 9, 8]

Перестановка (возврат опорного): [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Позиция опорного элемента после разделения: индекс 7

Опорный элемент: 9 (индекс 8)

Текущий подмассив: [9, 9]

Перестановка: [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Перестановка: [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Перестановка (возврат опорного): [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Позиция опорного элемента после разделения: индекс 9

Исходный массив: [4, 9, 1, 5, 4, 8, 3, 7, 9, 4]

Отсортированный массив: [1, 3, 4, 4, 4, 5, 7, 8, 9, 9]

Количество перестановок: 19

Массив отсортирован правильно

Нажмите Enter для продолжения...

3. Результат сортировки второго массива

СОРТИРОВКА НАБОРА 2

Исходный массив: [7, 2, 1, 6, 8, 5, 3, 4]

НАЧАЛО ПРОЦЕССА СОРТИРОВКИ:

Перестановка для выбора медианы: [6, 2, 1, 7, 8, 5, 3, 4]

Перестановка для выбора медианы: [4, 2, 1, 7, 8, 5, 3, 6]

Перестановка для выбора медианы: [4, 2, 1, 6, 8, 5, 3, 7]

Опорный элемент: 6 (индекс 3)

Текущий подмассив: [4, 2, 1, 6, 8, 5, 3, 7]

Перестановка: [4, 2, 1, 7, 8, 5, 3, 6]

Перестановка: [4, 2, 1, 3, 8, 5, 7, 6]

Перестановка: [4, 2, 1, 3, 5, 8, 7, 6]

Перестановка (возврат опорного): [4, 2, 1, 3, 5, 6, 7, 8]

Позиция опорного элемента после разделения: индекс 5

Перестановка для выбора медианы: [1, 2, 4, 3, 5, 6, 7, 8]

Опорный элемент: 4 (индекс 2)

Текущий подмассив: [1, 2, 4, 3, 5]

Перестановка: [1, 2, 5, 3, 4, 6, 7, 8]

Перестановка: [1, 2, 3, 5, 4, 6, 7, 8]

Перестановка (возврат опорного): [1, 2, 3, 4, 5, 6, 7, 8]

Позиция опорного элемента после разделения: индекс 3

Опорный элемент: 2 (индекс 1)

Текущий подмассив: [1, 2, 3]

Перестановка: [1, 3, 2, 4, 5, 6, 7, 8]

Перестановка (возврат опорного): [1, 2, 3, 4, 5, 6, 7, 8]

Позиция опорного элемента после разделения: индекс 1

Опорный элемент: 7 (индекс 6)

Текущий подмассив: [7, 8]

Перестановка: [1, 2, 3, 4, 5, 6, 8, 7]

Перестановка (возврат опорного): [1, 2, 3, 4, 5, 6, 7, 8]

Позиция опорного элемента после разделения: индекс 6

Исходный массив: [7, 2, 1, 6, 8, 5, 3, 4]

Отсортированный массив: [1, 2, 3, 4, 5, 6, 7, 8]

Количество перестановок: 15

Массив отсортирован правильно

Нажмите Enter для продолжения...

4. Результат сортировки третьего массива

СОРТИРОВКА НАБОРА 3

Исходный массив: [10, 80, 30, 90, 40, 50, 70]

НАЧАЛО ПРОЦЕССА СОРТИРОВКИ:

Перестановка для выбора медианы: [10, 80, 30, 70, 40, 50, 90]

Опорный элемент: 70 (индекс 3)

Текущий подмассив: [10, 80, 30, 70, 40, 50, 90]

Перестановка: [10, 80, 30, 90, 40, 50, 70]

Перестановка: [10, 50, 30, 90, 40, 80, 70]

Перестановка: [10, 50, 30, 40, 90, 80, 70]

Перестановка (возврат опорного): [10, 50, 30, 40, 70, 80, 90]

Позиция опорного элемента после разделения: индекс 4

Перестановка для выбора медианы: [10, 40, 30, 50, 70, 80, 90]

Опорный элемент: 40 (индекс 1)

Текущий подмассив: [10, 40, 30, 50]

Перестановка: [10, 50, 30, 40, 70, 80, 90]

Перестановка: [10, 30, 50, 40, 70, 80, 90]

Перестановка (возврат опорного): [10, 30, 40, 50, 70, 80, 90]

Позиция опорного элемента после разделения: индекс 2

Опорный элемент: 10 (индекс 0)

Текущий подмассив: [10, 30]

Перестановка: [30, 10, 40, 50, 70, 80, 90]

Перестановка (возврат опорного): [10, 30, 40, 50, 70, 80, 90]

Позиция опорного элемента после разделения: индекс 0

Опорный элемент: 80 (индекс 5)

Текущий подмассив: [80, 90]

Перестановка: [10, 30, 40, 50, 70, 90, 80]

Перестановка (возврат опорного): [10, 30, 40, 50, 70, 80, 90]

Позиция опорного элемента после разделения: индекс 5

Исходный массив: [10, 80, 30, 90, 40, 50, 70]

Отсортированный массив: [10, 30, 40, 50, 70, 80, 90]

Количество перестановок: 13

Массив отсортирован правильно

Вывод:

При помощи данной лабораторной работы, я научился реализовывать быструю сортировку