

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

**«Ижевский государственный технический университет имени
М.Т.Калашникова»**

(ФГБОУ ВПО ИжГТУ имени М.Т.Калашникова)

Лабораторная работа №1

по дисциплине «Алгоритмы и структуры данных»

Алгоритмы сортировки

Выполнил: студент гр. Б01-191-1зт

Шайхиев А.Ф.

Проверил(а): _____

Еланцев М.О.

Постановка задачи:

Реализовать два алгоритма сортировки на любом языке программирования целочисленного массива, согласно варианту(2 вариант) (пузырьковая сортировка и сортировка слиянием) .

1. Пузырьковая сортировка:

Это простой алгоритм сортировки. Для понимания и реализации этот алгоритм — простейший, но эффективен он лишь для небольших массивов.

Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов. Проходы по массиву повторяются тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован.

Исходный код:

```
#сортировка пузырьком
from random import randint

N = 5
a = []
for i in range(N):
    a.append(randint(1, 99))
print("Начальный массив:")
print(a)

print("Соритровка массива:")
for i in range(N-1):
    for j in range(N-i-1):
        if a[j] > a[j+1]:
            a[j], a[j+1] = a[j+1], a[j]
    print(a)

print("Отсортированный массив:")
print(a)
```

Результат выполнения алгоритма:

```
Начальный массив:
[45, 62, 90, 96, 53]
Соритровка массива:
[45, 62, 90, 53, 96]
[45, 62, 53, 90, 96]
[45, 53, 62, 90, 96]
[45, 53, 62, 90, 96]
Отсортированный массив:
[45, 53, 62, 90, 96]
```

На вход поступает массив из 5 элементов, нужно отсортировать его в порядке возрастания с помощью пузырьковой сортировки.

Сначала сравнивается 1ый и 2ой элементы массива, если условие сравнения выполняется, то меняем элементы местами, затем 2ой и 3ий элемент и так далее. В результате выполнения первого цикла на последнее место в массиве встанет максимальный элемент, затем цикл выполняется снова с первого элемента, пока все элементы массива не встанут на свои места.

2.Сортировка слиянием:

Сортировка слиянием реализуется рекурсивно в виде двух этапов: разбиение и слияние.

Задача разбивается на подзадачи меньшего размера, которые решаются по отдельности, после чего их решения комбинируются для получения решения исходной задачи. Конкретно процедуру сортировки слиянием можно описать следующим образом:

1. Если в рассматриваемом массиве один элемент, то он уже отсортирован — алгоритм завершает работу.
2. Иначе массив разбивается на две части, которые сортируются рекурсивно.
3. После сортировки двух частей массива к ним применяется процедура слияния, которая по двум отсортированным частям получает исходный отсортированный массив.

Исходный код:

```
def mergeSort(alist):
    print("Разделение элементов массива: ",alist)
    #деление массива,пока каждый элемент не станет отдельным элементом
    if len(alist)>1:
        mid = len(alist)//2
        lefthalf = alist[:mid]
        righthalf = alist[mid:]
        #рекурсивный вызов функции
        mergeSort(lefthalf)
        mergeSort(righthalf)

        i=0
        j=0
        k=0
        #перестановка в массиве
        while i<len(lefthalf) and j<len(righthalf):
            if lefthalf[i]<righthalf[j]:
                alist[k]=lefthalf[i]
                i=i+1
            else:
                alist[k]=righthalf[j]
                j=j+1
            k=k+1
        while i<len(lefthalf):
            alist[k]=lefthalf[i]
            i=i+1
            k=k+1
```

```

while j<len(righthalf):
    alist[k]=righthalf[j]
    j=j+1
    k=k+1
print("Слияние элементов массива: ",alist)

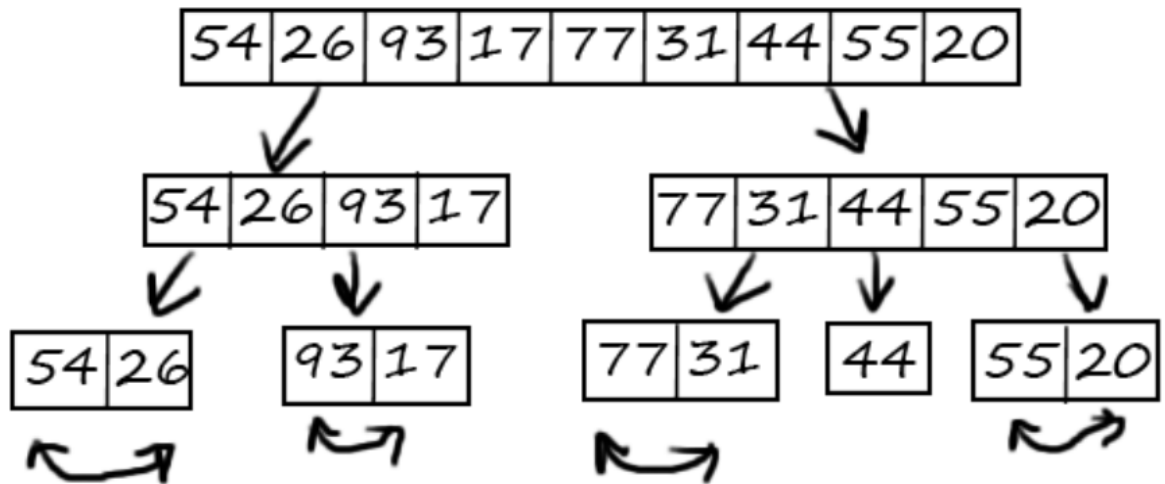
alist = [54,26,93,17,77,31,44,55,20]
mergeSort(alist)
print(alist)

```

Пример реализации алгоритма:

Дан исходный массив [54,26,93,17,77,31,44,55,20], нужно отсортировать его в порядке возрастания методом слияния.

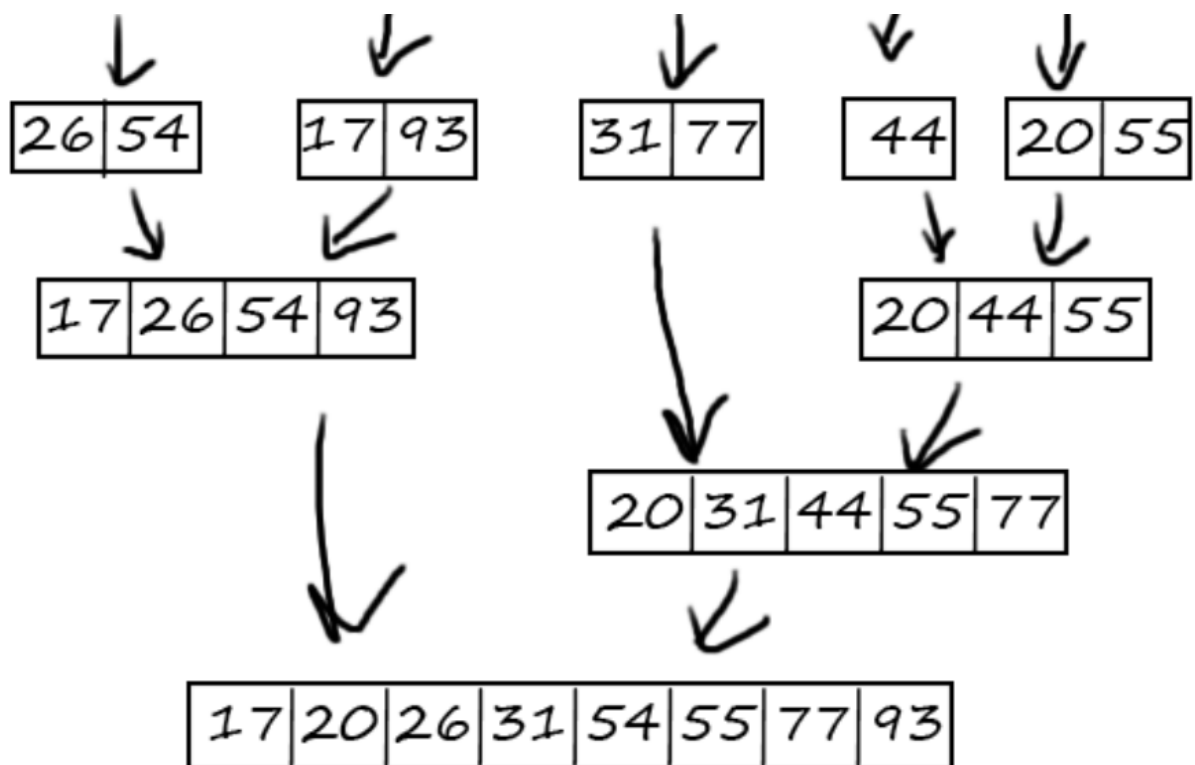
Для начала нам нужно разделить элементы массива на отдельные массивы, пока в них не останется 2 элемента, которые мы сможем сравнить между собой, если элемент один, то он уже отсортирован.



Пример работы программы:

```
Исходный массив [54, 26, 93, 17, 77, 31, 44, 55, 20]
Разделение элементов массива: [54, 26, 93, 17, 77, 31, 44, 55, 20]
Разделение элементов массива: [54, 26, 93, 17]
Разделение элементов массива: [54, 26]
Разделение элементов массива: [54]
Разделение элементов массива: [26]
Разделение элементов массива: [93, 17]
Разделение элементов массива: [93]
Разделение элементов массива: [17]
Разделение элементов массива: [77, 31, 44, 55, 20]
Разделение элементов массива: [77, 31]
Разделение элементов массива: [77]
Разделение элементов массива: [31]
Разделение элементов массива: [44, 55, 20]
Разделение элементов массива: [44]
Разделение элементов массива: [55, 20]
Разделение элементов массива: [55]
Разделение элементов массива: [20]
Отсортированный массив [17, 20, 26, 31, 44, 54, 55, 77, 93]
```

Затем мы сравниваем элементы между собой и меняем местами($54 > 26$ меняем местами 26 и 54), затем идем вверх по стеку и сравниваем отсортированные массивы и выполняем их слияние.



Пример работы программы:

```
Исходный массив [54, 26, 93, 17, 77, 31, 44, 55, 20]
Слияние элементов массива: [54]
Слияние элементов массива: [26]
Слияние элементов массива: [26, 54]
Слияние элементов массива: [93]
Слияние элементов массива: [17]
Слияние элементов массива: [17, 93]
Слияние элементов массива: [17, 26, 54, 93]
Слияние элементов массива: [77]
Слияние элементов массива: [31]
Слияние элементов массива: [31, 77]
Слияние элементов массива: [44]
Слияние элементов массива: [55]
Слияние элементов массива: [20]
Слияние элементов массива: [20, 55]
Слияние элементов массива: [20, 44, 55]
Слияние элементов массива: [20, 31, 44, 55, 77]
Слияние элементов массива: [17, 20, 26, 31, 44, 54, 55, 77, 93]
Отсортированный массив [17, 20, 26, 31, 44, 54, 55, 77, 93]
```

В результате работы алгоритма получаем отсортированный массив:

```
Исходный массив [54, 26, 93, 17, 77, 31, 44, 55, 20]
Отсортированный массив [17, 20, 26, 31, 44, 54, 55, 77, 93]
```