**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Інститут прикладного системного аналізу**

**Кафедра математичних методів системного аналізу**

**ЗВІТ**

про виконання Лабораторна робота № 2

дисципліни «Проектування та аналіз обчислювальних алгоритмів»

**Варіант №23**

**Виконав:**

Студент ІІ курсу

групи КА-86

Фролкін В. Ю.

Варіант № 23

**Перевірив:**

Севастьянов В.

**Київ** - **2020**

**Завдання:**

Реалізувати сортування: простого вибору та злиття

**Розв’язок:**

**Лістинг:**

def selection\_sort(array):

i, size = 0, len(array)

comps, swaps = 0, 0

while i < size:

min\_i, j = i, i + 1

while j < size:

comps += 1

if array[j] < array[min\_i]:

min\_i = j

j += 1

swaps += 1

array[i], array[min\_i] = array[min\_i], array[i]

i += 1

return array, comps, swaps

def merge(array, l, m, r):

comps, swaps = 0, 0

l\_copy = array[l:m + 1]

r\_copy = array[m + 1:r + 1]

l\_copy\_index = 0

r\_copy\_index = 0

sorted\_index = l

while l\_copy\_index < len(l\_copy) and r\_copy\_index < len(r\_copy):

comps += 1

if l\_copy[l\_copy\_index] <= r\_copy[r\_copy\_index]:

array[sorted\_index] = l\_copy[l\_copy\_index]

l\_copy\_index += 1

swaps += 1

else:

array[sorted\_index] = r\_copy[r\_copy\_index]

r\_copy\_index += 1

swaps += 1

sorted\_index += 1

while l\_copy\_index < len(l\_copy):

swaps += 1

array[sorted\_index] = l\_copy[l\_copy\_index]

l\_copy\_index = l\_copy\_index + 1

sorted\_index += 1

while r\_copy\_index < len(r\_copy):

swaps += 1

array[sorted\_index] = r\_copy[r\_copy\_index]

r\_copy\_index += 1

sorted\_index += 1

return comps, swaps

comps, swaps = 0, 0

def merge\_sort(array, left, right):

global comps, swaps

if left < right:

m = (left + right) // 2

merge\_sort(array, left, m)

merge\_sort(array, m + 1, right)

temp\_comps, temp\_swaps = merge(array, left, m, right)

comps += temp\_comps

swaps += temp\_swaps

return array, comps, swaps

**Результат роботи програми:**

*Selection sort:*

[32, 29, 81, 67, 51, 83, 44, 21, 49, 10, 41, 7, 71, 75, 8, 73, 47, 86, 2, 3, 80, 46, 20, 78, 16, 98, 14, 15, 82, 97, 12, 56, 48, 22, 50, 62, 77, 89, 61, 53, 42, 23, 40, 4, 17, 96, 68, 0, 92, 76, 91, 58, 45, 90, 70, 31, 25, 11, 94, 74, 5, 6, 85, 27, 52, 84, 55, 66, 65, 60, 69, 24, 13, 54, 9, 35, 57, 72, 59, 43, 38, 95, 93, 19, 33, 28, 87, 88, 79, 18, 64, 26, 37, 34, 1, 36, 99, 30, 63, 39]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]

*Merge sort:*

[20, 42, 10, 92, 51, 32, 43, 96, 53, 87, 13, 28, 22, 89, 93, 78, 55, 77, 79, 52, 65, 47, 31, 84, 83, 91, 39, 54, 40, 29, 25, 57, 98, 71, 82, 16, 73, 18, 3, 27, 97, 12, 45, 90, 24, 38, 95, 11, 66, 76, 19, 1, 0, 14, 5, 15, 46, 33, 59, 44, 99, 41, 80, 86, 7, 63, 23, 9, 56, 60, 26, 88, 2, 72, 58, 61, 4, 70, 68, 94, 64, 69, 8, 49, 81, 30, 62, 74, 48, 17, 85, 67, 50, 34, 75, 6, 37, 35, 21, 36]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]

**Теоретичні оцінки:**

****

**Аналіз результатів роботи:**

За результатами роботи алгоритмів видно, що сортування злиттям потребує менше порівнянь, але більше копіюваннь, через те, що елементи розділених масивів зливаються в один. В цілому сортування злиттям працює швидше, ніж сортування вибором.