

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 II курсу

групи КА-86

Фролкін В. Ю.

Варіант № 23

Перевірив:

Севастьянов В.

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:

До сортування:

[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]

Після сортування:

[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:

До сортування:

[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]

Після сортування:

[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:

Результати порівняння методів сортування										
N	Selection					Merge				
	К-ть копіювань (М)		К-ть порівнянь (С)		Час (Т)	К-ть копіювань (М)		К-ть порівнянь (С)		Час (Т)
	Теорет.	Експерим.	Теорет.	Експерим.		Теорет.	Експерим.	Теорет.	Експерим.	
100	99	99	4950	4950	0.00054	6644	672	-	548	0.00034
10000	9999	9999	49995000	49995000	4.9	13288	133616	-	120424	0.053
100000	99999	99999	4999950000	4999950000	549	16.610	1668928	-	1536461	0.056

Аналіз результатів роботи:

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