

Міністерство освіти і науки України  
Національний університет «Запорізька політехніка»

кафедра програмних засобів

Звіт  
з лабораторної роботи №3  
з дисципліни «Спортивне програмування»  
на тему : «Рекурсивні алгоритми»

**Виконала:**

студентка групи КНТ – 132

Мілена Кочева

**Прийняв:**

Старший викладач

Сергій ЛЕОЩЕНКО

2023

**Мета роботи:** вивчити основні можливості та принципи роботи рекурсивних алгоритмів.

**Хід роботи:**

*Завдання №1*

3.2.4 Впорядкувати за не зростанням 5 чисел за 7 операцій порівняння.

**Код:**

```
def mergeSort(arr):  
    if len(arr) > 1:  
        mid = len(arr) // 2  
        leftHalf = arr[:mid]  
        rightHalf = arr[mid:]  
  
        mergeSort(leftHalf)  
        mergeSort(rightHalf)  
  
        i = j = k = 0  
  
        while i < len(leftHalf) and j < len(rightHalf):  
            if leftHalf[i] >= rightHalf[j]:  
                arr[k] = leftHalf[i]  
                i += 1  
            else:  
                arr[k] = rightHalf[j]  
                j += 1  
            k += 1  
  
        while i < len(leftHalf):  
            arr[k] = leftHalf[i]  
            i += 1  
            k += 1
```

```

        while j < len(rightHalf):
            arr[k] = rightHalf[j]
            j += 1
            k += 1

def main():
    numbers = []

    for i in range(5):
        num = int(input(f"Введіть число {i + 1}: "))
        numbers.append(num)

    mergeSort(numbers)

    print("Відсортований масив за не зростанням:", numbers)

if __name__ == "__main__":
    main()

```

### Скріншот:

```

"C:\Program Files\Python3115\python.exe" D:\1_НУЗП\Спортивне\Lab_3\main.py
Введіть число 1: 5
Введіть число 2: 3
Введіть число 3: 1
Введіть число 4: 7
Введіть число 5: 9
Відсортований масив за не зростанням: [9, 7, 5, 3, 1]

Process finished with exit code 0
|

```

## Завдання №2

Є  $N$  осіб і цілі числа  $A_1, \dots, A_N$ ; людину  $i$  необхідно познайомити з  $A_i$  людьми.  
Чи можна це зробити?

### Код:

```
def canIntroduceAllPeople(N, A):  
    graph = {i: [] for i in range(1, N + 1)}  
  
    for i, ai in enumerate(A, start=1):  
        if ai > N - 1:  
            return False  
        for j in range(1, ai + 1):  
            if i + j <= N:  
                graph[i].append(i + j)  
  
    visited = set()  
  
    def dfs(node):  
        visited.add(node)  
        for neighbor in graph[node]:  
            if neighbor not in visited:  
                dfs(neighbor)  
  
    dfs(1)  
  
    return len(visited) == N  
  
N = 5  
A = [2, 1, 2, 1, 1]  
  
result = canIntroduceAllPeople(N, A)  
print(result)
```

### Скріншот:

```
"C:\Program Files\Python3115\python.exe" D:\1_НУЗП\Спортивне\Lab_3\Task_2\main.py  
True  
  
Process finished with exit code 0
```

**Висновок:** вивчила основні можливості та принципи роботи рекурсивних алгоритмів.