A

Uniwersytet Zielonogórski
Instytut Sterowania i Systemów Informatycznych

Algorytmy i struktury danych

Protokół laboratorium 3: Sortowanie

Imię i nazwisko: **Petros Palamiotis**

Grupa: **14INFSP-B**

Data: **30.10.2022**

Other:
Kod:

```
tab = [9, 11, 2, 6, 18, 37, 1, 42]


class color:
    BLUE = '\033[94m'
    GREEN = '\033[92m'
    YELLOW = '\033[93m'
    RED = '\033[91m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'
    END = '\033[0m'

def message(ar, st):
    if st:
        print("Posortowana tablica")
    else:
        print("Nieposortowana tablica")
    print(ar, "\n")
    return

def tab_print(t, a, b, c_a, c_b):
    for g in range(len(tab)):
        if g == a:
            print("  " + c_a + str(t[a]) + color.END, end="")
        elif g == b:
            print("  " + c_b + str(t[b]) + color.END, end="")
        else:
            print("  " + str(t[g]), end="")
    print("\n")
    return
```

Zadanie 1
Kod:

```
def insert_sort(array):
    message(array, False)

    for step in range(1, len(array)):
        print("Step %d" % step)
        print("Set key to %d" % array[step])
        tab_print(array, step, step+1, color.YELLOW, color.END)
        key = array[step]

        j = step - 1

        while j >= 0 and key < array[j]:
            if array[j+1] != array[j]:
                print("Put %d to %d" % (array[j], array[j+1]))
                tab_print(array, j+1, j, color.RED, color.END)

                array[j + 1] = array[j]

                tab_print(array, j+1, j, color.GREEN, color.END)

            j = j - 1

        if key != array[j + 1]:
            print("Put %d to %d" % (key, array[j + 1]))
            tab_print(array, j+1, step, color.RED, color.END)

            array[j + 1] = key

            tab_print(array, j+1, step, color.GREEN, color.END)

    message(array, True)

    return
```

Zadanie 2.
Kod:

```
def sort_choose(array, size):
    message(array, False)

    for step in range(size):
        index = step

        for i in range(step + 1, size):

            if array[i] < array[index]:
                index = i

        if array[step] != array[index]:
            print("Step %d\n" % (step + 1))
            print("Swap %d with %d\n" % (array[step], array[index]))
            tab_print(array, step, index, color.RED, color.GREEN)

        (array[step], array[index]) = (array[index], array[step])

        if array[step] != array[index]:
            tab_print(array, step, index, color.GREEN, color.RED)

    message(array, True)
    return
```

Zadanie 3.
Kod:

```
def bubble_sort(arr):
    message(arr, False)

    n = len(arr)

    for i in range(n):
        print("Step %d\n" % (i+1))

        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                print("Swap %d and %d" % (arr[j], arr[j+1]))
                tab_print(arr, j, (j+1), color.RED, color.GREEN)

                arr[j], arr[j+1] = arr[j+1], arr[j]

                tab_print(arr, j, (j+1), color.GREEN, color.RED)

    message(arr, True)

    return
```

Zadanie 4.
Kod:

```
def part(array, low, high):
    pivot = array[high]
    i = low - 1

    for j in range(low, high):
        if array[j] <= pivot:
            i = i + 1

            if array[i] != array[j]:
                print("Swap %d and %d\n" % (array[i], array[j]))
                tab_print(array, i, j, color.RED, color.GREEN)

            (array[i], array[j]) = (array[j], array[i])

            if array[i] != array[j]:
                tab_print(array, i, j, color.GREEN, color.RED)

    if array[i + 1] != array[high]:
        print("Swap %d and %d\n" % (array[i + 1], array[high]))
        tab_print(array, (i + 1), high, color.RED, color.GREEN)
    (array[i + 1], array[high]) = (array[high], array[i + 1])

    if array[i + 1] != array[high]:
        tab_print(array, (i + 1), (high), color.GREEN, color.RED)

    return i + 1


def quick_sort(array, low, high):
    if low < high:
        pi = part(array, low, high)

        quick_sort(array, low, pi - 1)

        quick_sort(array, pi + 1, high)

    return
```

Main
Kod:

```
def main():
    tab = [9, 11, 2, 6, 18, 37, 1, 42]

    print("### Zadanie 1 - Sortowanie poprzez wstawianie ###\n")
    insert_sort(tab)

    tab = [9, 11, 2, 6, 18, 37, 1, 42]

    print("### Zadanie 2 - Sortowanie poprzez wybieranie ###\n")
    sort_choose(tab, len(tab))

    tab = [9, 11, 2, 6, 18, 37, 1, 42]

    print("### Zadanie 3 - Sortowanie bąbelkowe ###\n")
    bubble_sort(tab)

    tab = [9, 11, 2, 6, 18, 37, 1, 42]

    print("### Zadanie 4 - Sortowanie Quicksort ###\n")
    message(tab, False)

    quick_sort(tab, 0, (len(tab) - 1))

    message(tab, True)

    return

if __name__ == '__main__':
    main()
```

Result:

```
### Zadanie 1 - Sortowanie poprzez wstawianie ###

Nieposortowana tablica
[9, 11, 2, 6, 18, 37, 1, 42]

Step 1
Set key to 11
  9  11  2  6  18  37  1  42

Step 2
Set key to 2
  9  11  2  6  18  37  1  42

Put 11 to 2
  9  11  2  6  18  37  1  42

  9  11  11  6  18  37  1  42

Put 9 to 11
  9  11  11  6  18  37  1  42

  9  9  11  6  18  37  1  42

Put 2 to 9
  9  9  11  6  18  37  1  42

  2  9  11  6  18  37  1  42

Step 3
Set key to 6
  2  9  11  6  18  37  1  42

Put 11 to 6
  2  9  11  6  18  37  1  42

  2  9  11  11  18  37  1  42

Put 9 to 11
  2  9  11  11  18  37  1  42

  2  9  9  11  18  37  1  42

Put 6 to 9
  2  9  9  11  18  37  1  42

  2  6  9  11  18  37  1  42

Step 4
Set key to 18
  2  6  9  11  18  37  1  42

Step 5
Set key to 37
  2  6  9  11  18  37  1  42

Step 6
Set key to 1
  2  6  9  11  18  37  1  42

Put 37 to 1
  2  6  9  11  18  37  1  42
  2   6   9   11   18   37   37   42

Put 18 to 37
  2   6   9   11   18   37   37   42

  2   6   9   11   18   18   37   42

Put 11 to 18
  2   6   9   11   18   18   37   42

  2   6   9   11   11   18   37   42

Put 9 to 11
  2   6   9   11   11   18   37   42

  2   6   9   9   11   18   37   42

Put 6 to 9
  2   6   9   9   11   18   37   42

  2   6   6   9   11   18   37   42

Put 2 to 6
  2   6   6   9   11   18   37   42

  2   2   6   9   11   18   37   42

Put 1 to 2
  2   2   6   9   11   18   37   42

  1   2   6   9   11   18   37   42

Step 7
Set key to 42
  1   2   6   9   11   18   37   42

Posortowana tablica
[1, 2, 6, 9, 11, 18, 37, 42]

### Zadanie 2 - Sortowanie poprzez wybieranie ###

Nieposortowana tablica
[9, 11, 2, 6, 18, 37, 1, 42]

Step 1

Swap 9 with 1

  9   11   2   6   18   37   1   42

  1   11   2   6   18   37   9   42

Step 2

Swap 11 with 2

  1   11   2   6   18   37   9   42

  1   2   11   6   18   37   9   42

Step 3

Swap 11 with 6
```

```
  1  2  11  6  18  37  9  42

  1  2  6  11  18  37  9  42

Step 4

Swap 11 with 9

  1  2  6  11  18  37  9  42

  1  2  6  9  18  37  11  42

Step 5

Swap 18 with 11

  1  2  6  9  18  37  11  42

  1  2  6  9  11  37  18  42

Step 6

Swap 37 with 18

  1  2  6  9  11  37  18  42

  1  2  6  9  11  18  37  42

Posortowana tablica
[1, 2, 6, 9, 11, 18, 37, 42]

### Zadanie 3 - Sortowanie bąbelkowe ###

Nieposortowana tablica
[9, 11, 2, 6, 18, 37, 1, 42]

Step 1

Swap 11 and 2
  9  11  2  6  18  37  1  42

  9  2  11  6  18  37  1  42

Swap 11 and 6
  9  2  11  6  18  37  1  42

  9  2  6  11  18  37  1  42

Swap 37 and 1
  9  2  6  11  18  37  1  42

  9  2  6  11  18  1  37  42

Step 2

Swap 9 and 2
  9  2  6  11  18  1  37  42

  2  9  6  11  18  1  37  42

Swap 9 and 6
  2  9  6  11  18  1  37  42
  2  6  9  11  18  1  37  42

Swap 18 and 1
  2  6  9  11  18  1  37  42

  2  6  9  11  1  18  37  42

Step 3

Swap 11 and 1
  2  6  9  11  1  18  37  42

  2  6  9  1  11  18  37  42

Step 4

Swap 9 and 1
  2  6  9  1  11  18  37  42

  2  6  1  9  11  18  37  42

Step 5

Swap 6 and 1
  2  6  1  9  11  18  37  42

  2  1  6  9  11  18  37  42

Step 6

Swap 2 and 1
  2  1  6  9  11  18  37  42

  1  2  6  9  11  18  37  42

Step 7

Step 8

Posortowana tablica
[1, 2, 6, 9, 11, 18, 37, 42]

### Zadanie 4 - Sortowanie Quicksort ###

Nieposortowana tablica
[9, 11, 2, 6, 18, 37, 1, 42]

Swap 9 and 1

  9  11  2  6  18  37  1  42

  1  11  2  6  18  37  9  42

Swap 11 and 2

  1  11  2  6  18  37  9  42

  1  2  11  6  18  37  9  42

Swap 11 and 6

  1  2  11  6  18  37  9  42
  1   2   6   11   18   37   9   42

Swap 11 and 9

  1   2   6   11   18   37   9   42

  1   2   6   9   18   37   11   42

Swap 18 and 11

  1   2   6   9   18   37   11   42

  1   2   6   9   11   37   18   42

Swap 37 and 18

  1   2   6   9   11   37   18   42

  1   2   6   9   11   18   37   42

Posortowana tablica
[1, 2, 6, 9, 11, 18, 37, 42]
```