

ЗАТВЕРДЖУЮ

Завідувач кафедри управління проектами, інформаційних технологій та телекомунікацій, д.т.н., професор

Є. В. Мартин

«__» _____ 20__ р.

МЕТОДИЧНА РОЗРОБКА

для проведення практичного заняття з студентами (курсантами) 1-го курсу спеціальності 122 «Комп'ютерні науки» з дисципліни «Основи програмування та алгоритмізація»

Практичне заняття 4.8: «Реалізація алгоритму швидкого сортування»

Мета: здобути навички щодо реалізації алгоритму швидкого сортування.

Час: 2 годин

Місце заняття: комп'ютерна лабораторія

Матеріальне забезпечення:

- ПК з відповідним ПЗ та доступом до мережі Internet;
- Методичні рекомендації щодо виконання практичної роботи.

Після практичного заняття студенти (курсанти) повинні:

вміти: реалізовувати алгоритм швидкого сортування масивів чисел у вигляді програмного коду.

знати: особливості застосування методу швидкого сортування.

Література:

1. Віртуальне навчальне середовище ЛДУ БЖД «Віртуальний університет» [Електронний ресурс]. – Доступний з : <http://virt.ldubgd.edu.ua/course/view.php?id=1672>

План заняття:

1. Складання тестових завдань за темою попередньої лекції.
2. Реалізація алгоритму швидкого сортування мовою програмування Java.

1. Складання тестових завдань за темою попередньої лекції.

Для складання тестових завдань необхідно перейти у відповідний розділ дисципліни «Основи програмування та алгоритмізація» у Віртуальному навчальному середовищі (ННІ цивільного захисту/спеціальність «Комп'ютерні науки»/бакалавр/І курс/II семестр/Основи програмування та алгоритмізація/Тема 4. Алгоритми/Контрольна частина/Тестове завдання за темою 4.7.)

Слід звернути увагу, що час та період складання тестового завдання обмежено.

2. Реалізація алгоритму швидкого сортування мовою програмування Java.

Перед виконання практичного завдання слід повторити лекцію «Швидке сортування», та нагадати особливості роботи цього методу сортування. Реалізацію алгоритму слід проводити з дотриманням таких вказівок: в якості опорного елемента обирається останній елемент масиву; прохід масиву має виконуватись з початкового до останнього елемента.

Код реалізації алгоритму наведено нижче:

```
public class QuickSort {
    int partition (int arr[], int first, int last) {
        int pivot = arr[last];
        int i = (first - 1);
        for (int j = first; j < last; j++) {
            if (arr[j] <= pivot) {
                i++;

                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
        int temp = arr[i + 1];
        arr[i + 1] = arr[last];
        arr[last] = temp;

        return i + 1;
    }

    void sort(int arr[], int first, int last) {
        if (first < last) {

            int pi = partition(arr, first, last);
            sort(arr, first, pi - 1);
            sort(arr, pi + 1, last);
        }
    }
}

import java.util.Arrays;

public class MainQuickSort {

    public static void main(String args[])
    {
        int[] arr = new int[100];
        for (int i = 0; i < arr.length; i++) {
            arr[i] = ((int) (Math.random() * 101));
        }
        System.out.println("no sorted array");
        System.out.println(Arrays.toString(arr));

        QuickSort ob = new QuickSort ();
    }
}
```

```

        ob.sort(arr, 0, arr.length-1);

        System.out.println(«sorted array»);
        System.out.println(Arrays.toString(arr));
    }

}

```

Існує й інший варіант реалізації алгоритму швидкого сортування, де в якості опорного елемента обирається середній елемент масиву, а прохід масиву реалізується в два етапи: ліворуч від першого до опорного елемента; праворуч від останнього до опорного елемента. Приклад реалізації подібного методу наведено нижче:

```

import java.util.Arrays;

public class QuickSort {

    public static void quickSort(int[] arr, int first, int last) {
        if (arr.length == 0)
            return;

        if (first >= last)
            return;

        int middle = first + (last - first) / 2;
        int opora = arr[middle];
        int i = first, j = last;
        while (i <= j) {
            while (arr[i] < opora) {
                i++;
            }

            while (arr[j] > opora) {
                j--;
            }

            if (i <= j) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
                i++;
                j--;
            }
        }

        if (first < j)
            quickSort(arr, first, j);

        if (last > i)
            quickSort(arr, i, last);
    }

    public static void main(String[] args) {

```

```

        int[] x = { 8, 0, 4, 7, 3, 7, 10, 12, -3 };
        System.out.println("no sorted array");
        System.out.println(Arrays.toString(x));

        int low = 0;
        int high = x.length - 1;

        quickSort(x, low, high);
        System.out.println("sorted array");
        System.out.println(Arrays.toString(x));
    }
}

```

В наведених вище прикладах реалізації алгоритму швидкого сортування передбачено вибір в якості опорного конкретного елемента. Проте з лекції відомо, що найбільш ефективним є алгоритм рандомізованого швидкого сортування із випадковим вибором опорного елемента. Саме тому в якості практичного завдання слід переписати подані алгоритми із врахуванням можливості випадкового вибору опорного елемента та його розміщенням у останню комірку масиву для подальшої реалізації Quick Sort.

Отже, **умова завдання**: Створити масив з 100 елементів. Наповнити масив випадково генерованими числами у проміжку від 1 до 100. Реалізувати алгоритм швидкого сортування із випадковим (рандомізованим) вибором опорного елемента таким чином, щоб на виході роботи програми масив наповнений «рандомованими» значеннями був відсортований у порядку зростання.

Код написаної програми необхідно завантажити у віртуальне навчальне середовище до категорії «Реалізація алгоритму швидкого сортування». Слід зауважити, що період виконання завдання обмежений.

Методичну розробку розробив:
 Заступник начальника кафедри УПІТтаТ
 майор служби цивільного захисту

О.В. Придатко

Методична розробка обговорена на засіданні кафедри УПІТтаТ
 Протокол № ____ від " __ " _____ 20__ р.