

UNIVERSITATEA "OVIDIUS" CONSTANȚA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

RAPORT ASUPRA PRACTICII: 25.06-06.07.2018

Student:
Turea Remus-Cristian

Cuprins

1	Introducere	2
2	Activități planificate	3
3	25.06.2018	4
4	26.06.2018	5
5	27.06.2018	6
6	28.06.2018	7
7	29.06.2018	8
8	02.07.2018	9
9	03.07.2018	11
10	04.07.2018	13
11	05.07.2018	14
12	06.07.2018	15
13	Concluzii	16

Capitolul 1

Introducere

Acest raport cuprinde descrierea activității desfășurate la practică în cadrul Facultății de Matematică și Informatică în perioada 25.06-06.07.2018.

Capitolul 2

Activități planificate

- Luni, 25.06.2018
Aducerea la cunoștință a obiectivelor și cerințelor practicii la calculator.
- Marți, 26.06.2018
Am ales tema: "Algoritmul de sortare rapidă (quick sort)".
- Miercuri, 27.06.2018
Am lucrat în Latex și am început să scriu raportul.
- Joi, 28.06.2018
M-am documentat cu privire la tema aleasă - "Algoritmul de sortare rapidă".
- Vineri, 29.06.2018
Am făcut o scurtă introducere și descriere a algoritmului de sortare rapidă.
- Luni, 02.07.2018
Am căutat un pseudocod pentru algoritmul de sortare rapidă.
- Marți, 03.07.2018
Am scris codul în limbajul java cu ajutorul NetBeans.
- Miercuri, 04.07.2018
Am testat funcționalitatea codului după ce am corectat erorile.
- Joi, 05.07.2018
Am finalizat raportul în latex după ce am adăugat etapele parcurse în zilele precedente.
- Vineri, 06.07.2018
Prezentarea lucrărilor. Notarea finală a activității.

Capitolul 3

25.06.2018

Am desfășurat următoarele activități:

- Am identificat sursele pentru MikTeX, Git, SmartGit și BitBucket.
 - Am identificat sursele pentru MikTeX, Git, SmartGit și BitBucket.
 - Am instalat, configurat pe calculatorul de lucru aplicațiile necesare:
 - * MikTeX
 - * SmartGit
 - * Bitbucket
 - Am instalat, configurat pe calculatorul de lucru aplicațiile necesare:
 - * MikTeX
 - * SmartGit
 - * Bitbucket

Studierea obiectivelor și cerințelor față de practica de producție. Clarificarea situațiilor incerte.

Capitolul 4

26.06.2018

Am ales tema: "Algoritmul de sortare rapidă (quick sort)".

Capitolul 5

27.06.2018

Am studiat și lucrat în Latex și am început să scriu raportul.

Capitolul 6

28.06.2018

M-am documentat cu privire la tema aleasă - "Algoritmul de sortare rapidă".

Capitolul 7

29.06.2018

Sortarea rapidă este un algoritm de sortare care, pentru un sir de n elemente, are un timp de execuție $O(n^2)$, în cazul cel mai defavorabil. În ciuda acestei comportări proaste, în cazul cel mai defavorabil, algoritmul de sortare rapidă este deseori cea mai bună soluție practică, deoarece are o comportare medie remarcabilă: timpul său mediu de execuție este $O(n \lg n)$, și constanta ascunsă în formula $O(n \lg n)$ este destul de mică. Algoritmul are avantajul că sortează pe loc (în spațiul alocat sirului de intrare) și lucrează foarte bine chiar și într-un mediu de memorie virtuală. Algoritmul de sortare rapidă se bazează pe paradigma “divide și stăpânește”.

Capitolul 8

02.07.2018

Am găsit un pseudocod pentru tema selectată.

QUICKSORT(A, p, r)

1. dacă $p < r$ atunci
2. $q \leftarrow \text{PARTIȚIE}(A, p, r)$
3. QUICKSORT(A, p, q)
4. QUICKSORT(A, q + 1, r)

Pentru ordonarea întregului sir A, inițial se apelează Quicksort(A, 1, [A]). Cheia algoritmului este procedura Partiție, care rearanjează pe loc subsirul A[p..r].

PARTIȚIE(A, p, r)

1. $x \leftarrow A[p]$
2. $i \leftarrow p - 1$
3. $j \leftarrow r - 1$
4. cât timp ADEVĂRAT execută
5. repetă
6. $j \leftarrow j - 1$
7. pâna când $A[j] \leq x$
8. repetă
9. $i \leftarrow i + 1$

10. până când $A[i] \geq x$
11. dacă $i < j$ atunci
12. interschimbă $A[i] \leftrightarrow A[j]$
13. altfel
14. returnează j

Capitolul 9

03.07.2018

Am scris codul java în programul NetBeans:

```
class QSort
{
    int partitie(int x[], int min, int max)
    {
        int piv = x[max];
        int i = (min-1);
        for (int j=min; j <max; j++)
        {
            if (x[j] < piv)
            {
                i++;
                int aux = x[i];
                x[i] = x[j];
                x[j] = aux;
            }
        }

        int aux = x[i+1];
        x[i+1] = x[max];
        x[max] = aux;
        return i+1;
    }

    void sort(int x[], int min, int max)
    {
        if (min < max)
        {
            int pi = partitie(x, min, max);
            sort(x, min, pi-1);
            sort(x, pi+1, max);
        }
    }
}
```

```

    }
}

static void print(int x[])
{
    int n = x.length;
    for (int i=0; i <n; ++i)
        System.out.print(x[i]+" ");
    System.out.println();
}

public static void main(String args[])
{
    int x[] = {12, 17, 3, 6, 1, 26, 89, 45, 111, 67};
    int n = x.length;

    QSort obj = new QSort();
    obj.sort(x, 0, n-1);

    System.out.println("sirul sortat");
    print(x);
}
}

```

Capitolul 10

04.07.2018

Am testat funcționalitatea codului după ce am corectat erorile.

Capitolul 11

05.07.2018

Am finalizat raportul în latex după ce am adăugat etapele parcurse în zilele precedente.

Capitolul 12

06.07.2018

Prezentarea proiectului. Notarea finală a activității.

Capitolul 13

Concluzii

Am învățat să lucrez cu Latex și Git. În latex am învățat cum să structurez paginile, să scriu cu diacritice utilizand pachetele “`\usepackage{ucs}`” si “`\usepackage[english,romanian]{babel}`”, să alcătuiesc un cuprins, să adaug liste, să utilizez bibliografia.

M-am familiarizat cu modul de lucru al Githubului creandu-mi un cont si am realizat două depozite;în primul am introdus fisierele implementării algoritmului de sortare rapidă, iar în al doilea am introdus fisierele care conțin raportul de practică.

Am utilizat cărțile: [1] pentru familiarizarea cu LaTeX si [2] pentru a studia tema aleasă ”algoritmul de sortare rapidă”.

Bibliografie

- [1] Ludmila Malahova Alexandru Colesnicov. *Latex prin exemple*. 2001.
- [2] Ronald R. Rivest Thomas H. Cormen, Charles E. Leiserson. *Introducere în algoritmi*. 2000.