

Ministerul Educației a Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică

RAPORT

Lucrarea de laborator nr. 2

La disciplina: **Analiza și proiectarea algoritmilor**

A efectuat:
student a gr. TI-151 *Poseletchi Cristian*
A verificat:
lect. sup. *Veronica Bagrin*

Chișinău 2016

Lucrarea de Laborator № 2

Sarcina lucrarii:

1. De aplicat in cod algoritmele de sortare : Mergesort , Quicksort si Bubblesort.
2. De aflat numarul de iteratii la fiecare sortare.
3. De introdus timpul de executare in cod.
4. De construit tabel dupa timpul de executare si numarul de iteratii.
5. De construit un graf dupa tabel.

Codul programului:

```
#define _CRT_SECURE_NO_WARNINGS
#include<conio.h>
#include<iostream>
#include<time.h>
#include <vector>

using namespace std;
int it;
const int n = 100;
int tab[n];

// Quicksort .
int partition(int InitialTablou[], int top, int bottom)
{
    int x = InitialTablou[top];
    int i = top - 1;
    int j = bottom + 1;
    int temp;

    do
    {
        do
        {
            j--;
        } while (x < InitialTablou[j]);

        do
        {
            i++;
        } while (x > InitialTablou[i]);

        if (i < j)
        {
            temp = InitialTablou[i];
            InitialTablou[i] =
                InitialTablou[j];
            InitialTablou[j] = temp;
        }
    } while (i < j);

    it = i + j;
    return j;
}
```

```

void quicksort(int InitialTablou[], int top, int
bottom)
{
    int middle;

    if (top < bottom)
    {
        middle = partition(InitialTablou, top,
bottom);
        quicksort(InitialTablou, top, middle);
        quicksort(InitialTablou, middle + 1,
bottom);
    }
}

```

```

void AfisareTablou(int InitialTablou[], int size)
{
    int i;

    for (i = 0; i < size; i++)
    {
        cout << InitialTablou[i] << ' ';

    }
    cout << "\n\n";
}

```

```

int main()

```

```

{
    int initialTablou[1000];

    int n = 50;

    rand();

    int marime = n;
    for (int i = 0; i < marime; i++)
        tab[i] = rand() % 1000 - 100;

    cout << "Tablou nesortat:"
<< endl;

    AfisareTablou(tab, marime);

    clock_t start, end;

    start = clock();

    quicksort(tab, 0, marime-1);

    cout << endl;
    cout << endl;

    cout << "Tablou sortat prin
metoda quiksort este:" << endl;

    AfisareTablou(tab, marime );

    cout << endl;

    printf("\nTimpul de executie
al algoritmului Quicksort este:: %.8f ", ((double)
(clock() - start)) / CLOCKS_PER_SEC);

    end = clock();

    cout << endl;

    cout<<"Numarul de iteratii :
" <<it<< endl;

    _getch();

    return 0;
}

```

```

}

#define _CRT_SECURE_NO_WARNINGS
#include<conio.h>
#include<iostream>
#include<time.h>

using namespace std;

const int n = 100;
int tab[n];
int it;

//MergeSort
void MS( int aux1,int aux2,int aux3, int aux4)
{
    int i, j;
    for (j = aux1;j <= aux2;j++)
        for (i = aux3;i <= aux4;i++)
            if (tab[i]<tab[j])
            {
                tab[i] += tab[j];
                tab[j] = tab[i] -
tab[j];

                tab[i] -= tab[j];
                it = i + j;
            }
}

void AF(int tab[], int size)

```

```

{
    int i;

    for (i = 0; i < size; i++)
    {
        cout << tab[i] << ' ';

    }

    cout << "\n\n";
}

void insert(int i, int j)
{
    if (tab[i]>tab[j])
    {
        tab[i] += tab[j];
        tab[j] = tab[i] - tab[j];
        tab[i] -= tab[j];
    }
}

int MergeSort(int i, int j)
{
    if (j - i <= 1)insert(i, j);
    else
    {
        MergeSort(i, (i + j) / 2);
        MergeSort(1 + (i + j) / 2, j);
        MS(i, (i + j) / 2, 1 + (i + j) / 2, j);
    }
}

```

```

    }

    return tab;
}

int main()
{
    int initialTablou[10000];
    int n = 50, array[10000];

    rand();

    int marime = n;
    for (int i = 0; i < marime; i++)
        tab[i] = rand() % 1000 - 100;

    {
        clock_t start, end;\
        start = clock();
        for (int i = 0; i < n; i++)
            MergeSort(0, n - 1);
        start = clock();
        cout << "Tabloul sortat prin
metoda MergeSort este:" << endl;
        MergeSort(0, n - 1);
        AF(tab, marime);
        cout << "Numarul de iteratii
: "<<it << endl;

        cout << endl;

        printf("Timpul de executie al
algoritmului MergeSort este: %.8f\n", ((double)
(clock() - start)) / CLOCKS_PER_SEC);
    }
}

```

```

    _getch();
    return 0;
}

#define _CRT_SECURE_NO_WARNINGS
#include<conio.h>
#include<iostream>
#include<time.h>
#include <vector>

using namespace std;
int it;
const int n = 100;
int tab[n];

void bubble_sort(int iarr[], int num) {
    int i, j=0, k, temp;
    bool swapped = true;

    while (swapped) {
        swapped = false;
        j++;
    }
}

```

```

for (i = 0; i < num-j; i++) {
    if (iarr[i] > iarr[i + 1]) {
        temp = iarr[i];
        iarr[i] = iarr[i + 1];
        iarr[i + 1] = temp;
        swapped = true;
    }
}

cout << endl;

it = ((i)*(j));

}

void AfisareTablou(int iarr[], int size)
{
    int i;

    for (i = 0; i < size; i++)
    {
        cout << iarr[i] << ' ';
    }
}

cout << "\n\n";
}

int main()
{
    int initialTablou[1000];
    int n = 50;
    rand();

    int marime = n;
    for (int i = 0; i < marime; i++)
        tab[i] = rand() % 1000 - 100;

    cout << "Tablou nesortat:" << endl;
    AfisareTablou(tab, marime);
    clock_t start, end;
    start = clock();
    bubble_sort(tab, marime);
    cout << endl;
    cout << endl;
    cout << "Tablou sortat prin metoda bubble
sort este:" << endl;
    AfisareTablou(tab, marime);

    cout << endl;

    printf("\nTimpul de executie al algoritmului
bubble sort este:: %.8f ", ((double)(clock() -
start)) / CLOCKS_PER_SEC);

    end = clock();
}

```

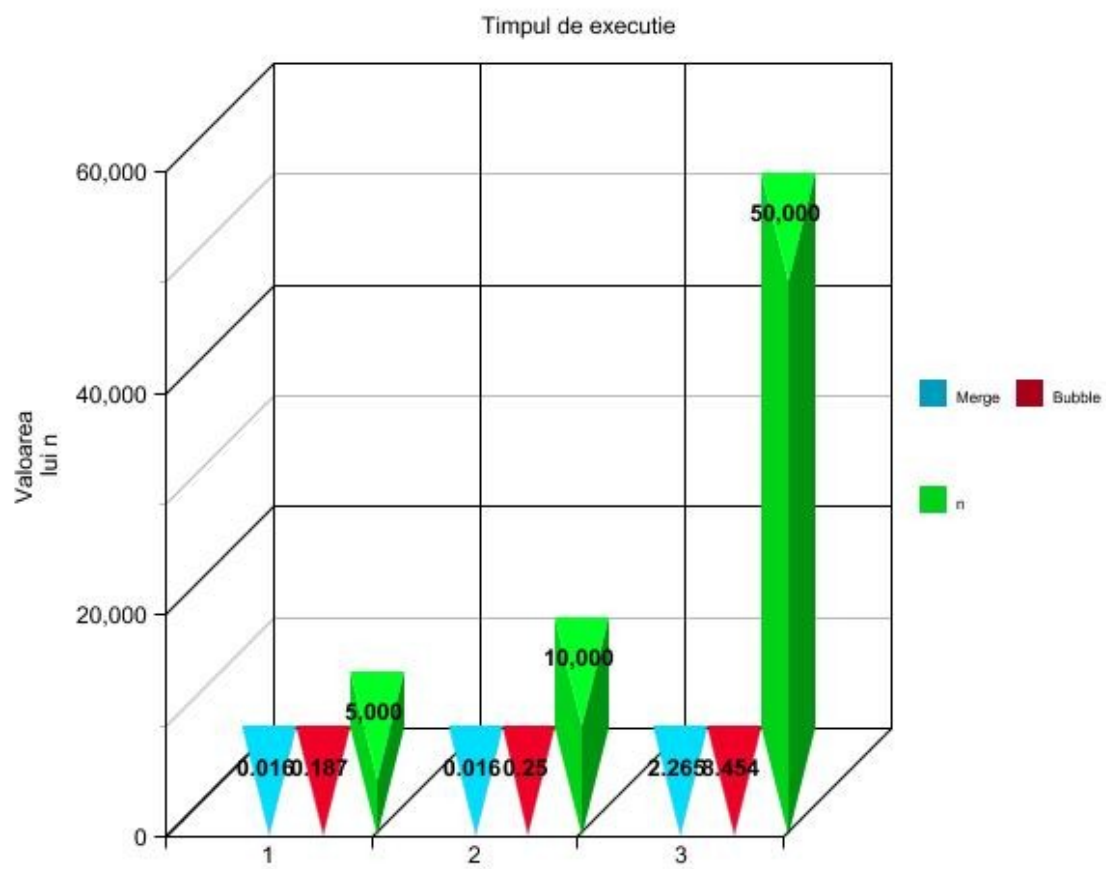
```
cout << endl;

cout << "Numarul de iteratii : " << it << endl;

_getch();

return 0;

}
```



Rezultatul afisarii

Metoda QuickSort

N=10


```
c:\users\aily\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605

Tabloul sortat prin metoda quiksort este:
69 234 258 364 367 378 400 605 624 862

Timpul de executie al algoritmului Quicksort este:: 0.00700000
Numarul de iteratii : 17
```

N=20

```
c:\users\aily\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605 45 181 727 861 391 895 842 727 336 291

Tabloul sortat prin metoda quiksort este:
45 69 181 234 258 291 336 364 367 378 391 400 605 624 727 727 842 861 862 895

Timpul de executie al algoritmului Quicksort este:: 0.01100000
Numarul de iteratii : 37
```

N=30

```
c:\users\aily\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605 45 181 727 861 391 895 842 727 336 291 504
4 802 53 192 282 321 616 618 795 347

Tabloul sortat prin metoda quiksort este:
45 53 69 181 192 234 258 282 291 321 336 347 364 367 378 391 400 504 605 616 618
624 727 727 795 802 842 861 862 895

Timpul de executie al algoritmului Quicksort este:: 0.01700000
Numarul de iteratii : 57
```

N=40

```
c:\users\aily\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605 45 181 727 861 391 895 842 727 336 291 504
4 802 53 192 282 321 616 618 795 347 626 671 438 769 812 567 199 -65 794 603

Tabloul sortat prin metoda quiksort este:
-65 45 53 69 181 192 199 234 258 282 291 321 336 347 364 367 378 391 400 438 504
567 603 605 616 618 624 626 671 727 727 769 794 795 802 812 842 861 862 895

Timpul de executie al algoritmului Quicksort este:: 0.02100000
Numarul de iteratii : 75
```

N=50

```
c:\users\aily\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605 45 181 727 861 391 895 842 727 336 291 504
4 802 53 192 282 321 616 618 795 347 626 671 438 769 812 567 199 -65 794 603 711
222 233 573 564 41 611 153 768 447

Tabloul sortat prin metoda quiksort este:
-65 41 45 53 69 153 181 192 199 222 233 234 258 282 291 321 336 347 364 367 378
391 400 438 447 504 564 567 573 603 605 611 616 618 624 626 671 711 727 727 768
769 794 795 802 812 842 861 862 895

Timpul de executie al algoritmului Quicksort este:: 0.02400000
Numarul de iteratii : 94
```

Metoda MergeSort:

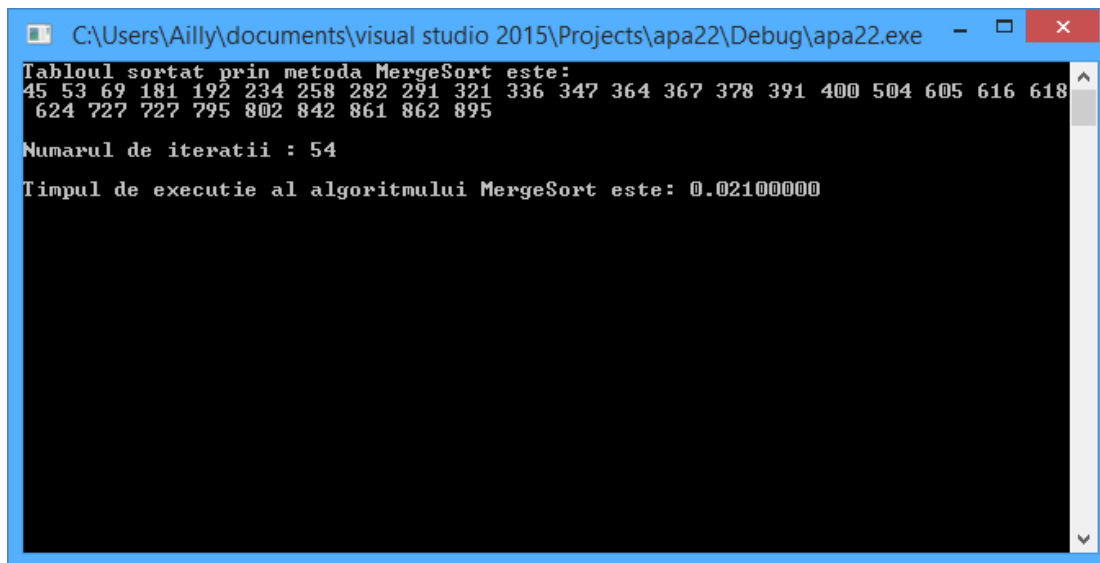
N=10

```
C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul sortat prin metoda MergeSort este:
69 234 258 364 367 378 400 605 624 862
Numarul de iteratii : 15
Timpul de executie al algoritmului MergeSort este: 0.01100000
```

N=20

```
C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul sortat prin metoda MergeSort este:
45 69 181 234 258 291 336 364 367 378 391 400 605 624 727 727 842 861 862 918
Numarul de iteratii : 35
Timpul de executie al algoritmului MergeSort este: 0.01500000
```

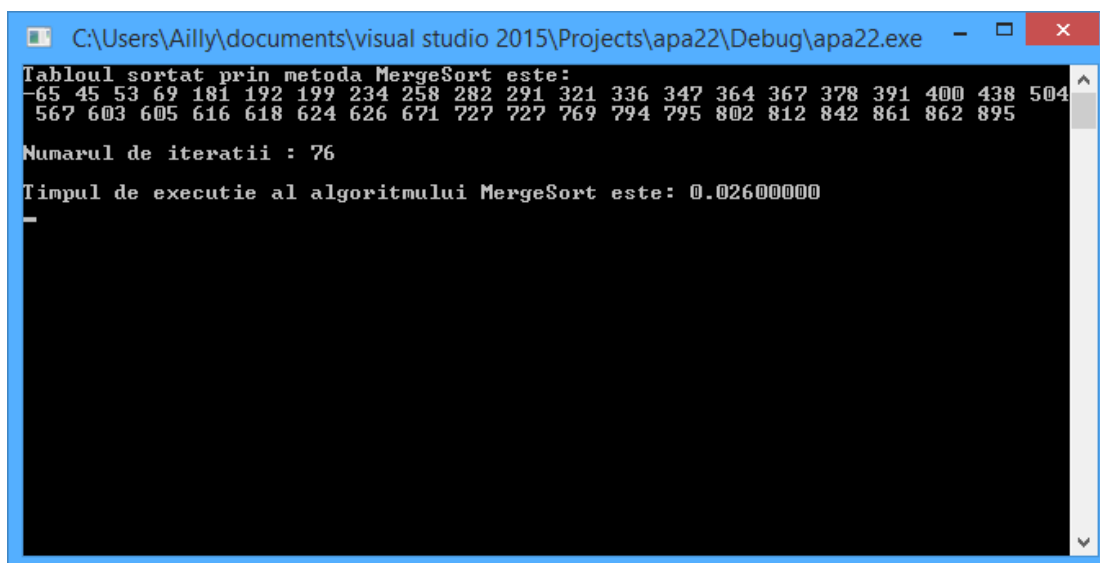
N=30



C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe

```
Tabloul sortat prin metoda MergeSort este:  
45 53 69 181 192 234 258 282 291 321 336 347 364 367 378 391 400 504 605 616 618  
624 727 727 795 802 842 861 862 895  
  
Numarul de iteratii : 54  
  
Timpul de executie al algoritmului MergeSort este: 0.02100000
```

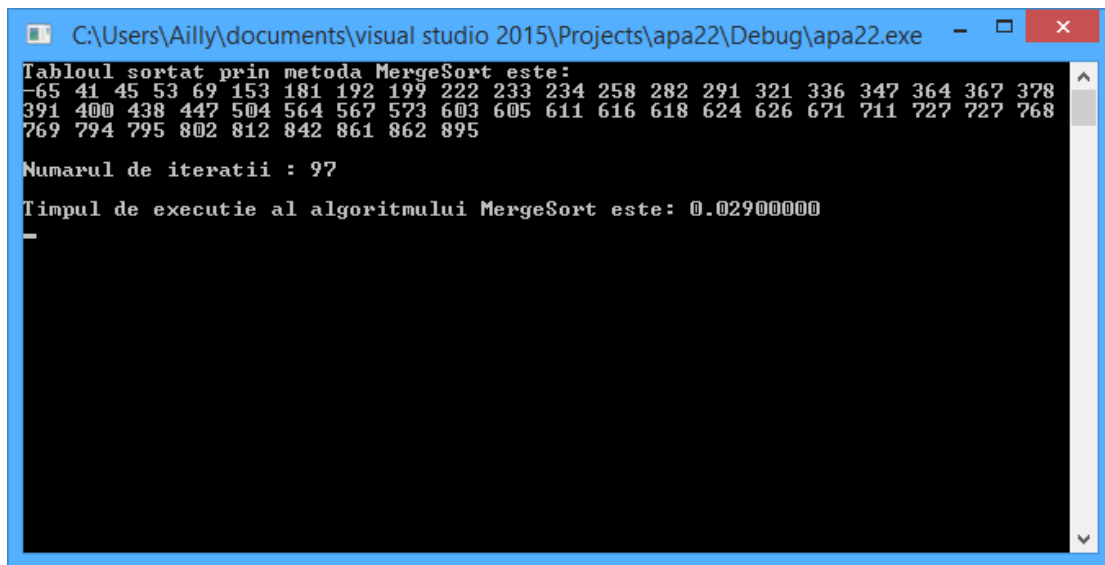
N=40



C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe

```
Tabloul sortat prin metoda MergeSort este:  
-65 45 53 69 181 192 199 234 258 282 291 321 336 347 364 367 378 391 400 438 504  
567 603 605 616 618 624 626 671 727 727 769 794 795 802 812 842 861 862 895  
  
Numarul de iteratii : 76  
  
Timpul de executie al algoritmului MergeSort este: 0.02600000  
-
```

N=50



```
C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul sortat prin metoda MergeSort este:
-65 41 45 53 69 153 181 192 199 222 233 234 258 282 291 321 336 347 364 367 378
391 400 438 447 504 564 567 573 603 605 611 616 618 624 626 671 711 727 727 768
769 794 795 802 812 842 861 862 895
Numarul de iteratii : 97
Timpul de executie al algoritmului MergeSort este: 0.02900000
```

Metoda BubbleSort

N=10

```
C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe - □ ×

Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605

Tabloul sortat prin metoda bubble sort este:
69 234 258 364 367 378 400 605 624 862

Timpul de executie al algoritmului bubble sort este:: 0.00900000
Numarul de iteratii : 24
-
```

N=20

```
C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe - □ ×

Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605 45 181 727 861 391 895 842 727 336 291

Tabloul sortat prin metoda bubble sort este:
45 69 181 234 258 291 336 364 367 378 391 400 605 624 727 727 842 861 862 895

Timpul de executie al algoritmului bubble sort este:: 0.01400000
Numarul de iteratii : 75
-
```

N=30

```
C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605 45 181 727 861 391 895 842 727 336 291 504
802 53 192 282 321 616 618 795 347

Tabloul sortat prin metoda bubble sort este:
45 53 69 181 192 234 258 282 291 321 336 347 364 367 378 391 400 504 605 616 618
624 727 727 795 802 842 861 862 895

Timpul de executie al algoritmului bubble sort este:: 0.01800000
Numarul de iteratii : 176
```

N=40

```
C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe
Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605 45 181 727 861 391 895 842 727 336 291 504
802 53 192 282 321 616 618 795 347 626 671 438 769 812 567 199 -65 794 603

Tabloul sortat prin metoda bubble sort este:
-65 45 53 69 181 192 199 234 258 282 291 321 336 347 364 367 378 391 400 438 504
567 603 605 616 618 624 626 671 727 727 769 794 795 802 812 842 861 862 895

Timpul de executie al algoritmului bubble sort este:: 0.02200000
Numarul de iteratii : 152
```

N=50

```
C:\Users\Ailly\documents\visual studio 2015\Projects\apa22\Debug\apa22.exe - □ ×
Tabloul nesortat:
367 234 400 69 624 378 258 862 364 605 45 181 727 861 391 895 842 727 336 291 50
4 802 53 192 282 321 616 618 795 347 626 671 438 769 812 567 199 -65 794 603 711
222 233 573 564 41 611 153 768 447

Tabloul sortat prin metoda bubble sort este:
-65 41 45 53 69 153 181 192 199 222 233 234 258 282 291 321 336 347 364 367 378
391 400 438 447 504 564 567 573 603 605 611 616 618 624 626 671 711 727 727 768
769 794 795 802 812 842 861 862 895

Timpul de executie al algoritmului bubble sort este:: 0.02600000
Numarul de iteratii : 225
```

Concluzie:

Efectuînd lucrarea dată am implementat algoritmi de sortare, quicksort , mergesort si bubblesort, bazați pe analizind timpul de executie al algoritmilor dați. Acești algoritmi au un timp de execuție diferit, de aceea este important de a alege algoritmul cel mai eficient, adică algoritmul care are un timp de execuție mic, se observa :

1. Din cele analizate se vede ca quicksort este un algoritm mai rapid, la quick sort se efectueaza mai multe iteratii.
2. Dar se observa ca dupa analiza si executare cel mai incet lucreaza sortarea dupa bubblesort.