METHODOLOGY OF LEARNING SORTING ALGORITHMS WITH THE HELP OF VISUALIZATION OF SORTING PROCESS IN CONSOLE MODE

Sadullaeva Sh.A., Beknazarova S.S., Fozilov F.D.
(TUIT named after Muhammad Al-Khwarizmi.)

Abstract: This article describes in detail the process of implementing a sort using a bubble sort as an example. Sorting was also described in this article. Students studying sorting algorithms will be able to learn sorting algorithms and create their own sorting visualizer

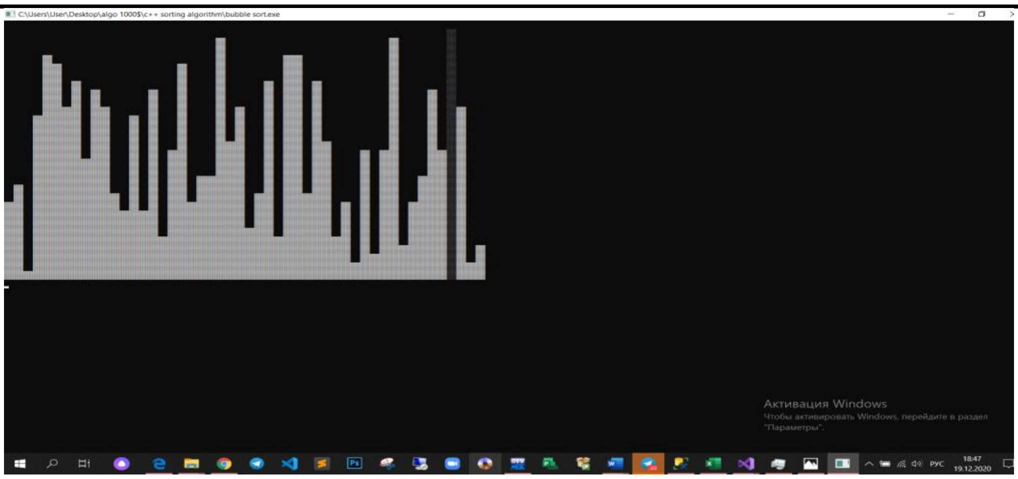
Bubble sort
Or sorting with simple exchanges. Immortal classic of the genre. The principle of action is simple: we go through the array from beginning to end, simultaneously swapping the places of unsorted adjacent elements. As a result of the first pass, the maximum element will "pop up" to the last place. Now we go through the unsorted part of the array again (from the first element to the penultimate one) and change the unsorted neighbors along the way. The second largest element will be in the penultimate place. Continuing in the same spirit, we will bypass the ever-decreasing unsorted part of the array, pushing the found maximums to the end

Sorting algorithm visualizer.
In order to create a visualizer program, you first need to understand how sorting itself works. After that, you need to create a program using this algorithm. The most important thing in each iteration of the sorting or process is to output the obtained result to the place in order to display the result at the end of the sort. The result will be a visualization of the entire sorting process. For visualization, we will use bars of length, which are equal to the length of the numbers in the array. The total number of bars is equal to the number of numbers in the array. So one bar represents one number in the array. During the iteration process, each bar will change its position and be displayed on the screen. No graphical environment will be used to implement this renderer. The program will be implemented in console mode, and at every moment of the iteration, the console will be cleared and display the bars anew.

Program code.
```
#include <iostream>
//connect the input and output library
#include <ctime>
```

```
//connect the time library to use the random input function
#include <stdlib.h>
using namespace std;
int main() // this is the main function where our program will run
{
        srand(time(NULL));
        // here we use the srand function in order to use the random numbers
        int a[50];
        // declaring the array for sorting it
        for(int i=0; i<50; i++)
        {
                a[i]=rand()%29+1;
                // here in loop we give the value for each member of the array randomly
from one to 30
        }
        cout<<endl;
        int b[30][50];
        // we declare b array, so we could visualize it in console
                for(int i=0; i<50; i++)
                {
                // here we use the bubble sort algorithm and visualize it in console mode
                        for(int j=0; j<49-i; j++)
                        {
                                if(a[j]>a[j+1])
                                {
                                        int temp=a[j];
                                        a[j]=a[j+1];
                                        a[j+1]=temp;
                                }
                                int k=j+1;
                                for(int i=0; i<30; i++)
                                {
                                        for(int j=0; j<50; j++)
                                        {
                                                if(a[j]>=(30-i))
                                                        if(j==k)
                                                                b[i][j]=2;
                                                        else
                                                                b[i][j]=1;
                                                else
                        b[i][j]=0;
                    }
                }
            // in each iteration we draw our array
                for(int i=0; i<30; i++)
                {
                    for(int j=0; j<50; j++)
                    {
                        if (b[i][j]==1)
                        {
                            cout<<char(178);
                            cout<<char(178);
                        }
                        else if(b[i][j]==2)
                        {
                            cout<<char(176);
                            cout<<char(176);
                        }
                        else
                        cout<<"  ";
                    }
                    cout<<endl;
                }
                system("cls");
//finally, we clear the console so we could draw it again with new iteration
            }
        }
}
```

Program result


C:\Users\User\Desktop\algo 1000$\c++ sorting algorithm\bubble sort.exe
Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".
РУС
18:47
19.12.2020

The implementation of such a visualizer helps the student to better understand the sorting algorithm itself. The teaching method is for the student to be able to create the sorting visualizer himself.