

Сортиране на масиви

Видове сортировки и приложение

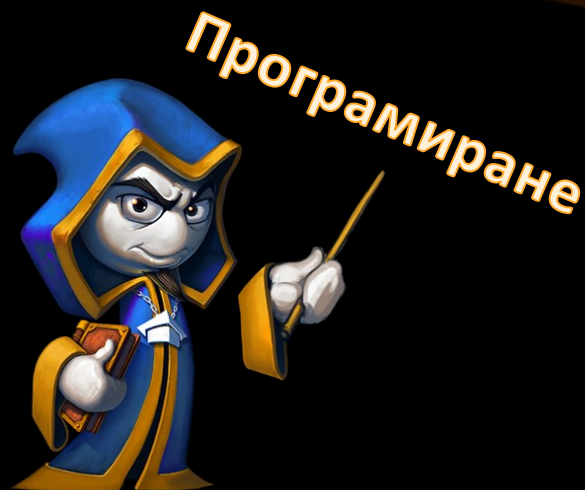


Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>

<https://github.com/BG-IT-Edu/School-Programming/tree/main/Courses/Applied-Programmer/Programming-Fundamentals>



2 4 -5 1 10



-5 1 2 4 10



Съдържание

1. Що е сортиране и свойствата му
2. Начини за сортиране
3. Някои известни методи на сортиране
4. Задачи



Що е сортиране?

- **Сортирането** на множество представлява **подреждане** на елементите му по даден **признак**
 - Подредбата е във **възходящ** или **низходящ** ред
- Възможно е подреждането да бъде направено по **няколко критерия**
 - Ако два елемента имат една и съща стойност по даден признак, се подреждат по **следващ признак**
 - Подредбата е по повече от един критерий, като има значение кой критерий е първи и кой втори

Характеристики на сортирането на множества

- Сортираме (подреждаме) множества за **по-бързо търсене** на елементи в него
- Основните особености на едно сортиране са:
 - **Сложността** (брой сравнения и размени на елементи)
 - **Използвани ресурси** (памет)
 - **Стабилност** (дали елементите се разместват по друг критерий, ако по критерия по който подреждаме са равни)
 - **Реализацията** на метода зависи от **структурата на паметта**, в която са записани данните

Някои известни методи на сортиране

- Метод на мехурчето (bubble sort)
- Пряка селекция (selection sort)
- Сортиране чрез вмъкване (insertion sort)
- Бърза сортировка (quicksort)

Метод на мехурчето

```
using System;
public class Program
{
    public static void Main()
    {
        int[] arr = new int[] { 2, 4, -5, 1, 10 };
        for (int i = 0; i < arr.Length - 1; i++)
        {
            for (int j = 0; j < arr.Length - 1 - i; j++)
            {
                if (arr[j] > arr[j + 1])
                {
                    int swapVar = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = swapVar;
                }
            }
            for (int i = 0; i < arr.Length; i++)
                Console.Write(arr[i] + " ");
        }
    }
}
```

Масив от 5 елемента

N-1 Пъти правим

обхождане от първия до
последния елемент и
сравняваме два съседни



елемента

Сложност $O(n^2)$

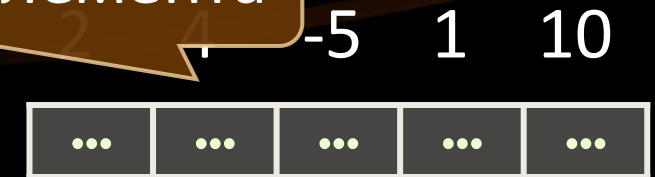
Ако условието за
размяна е на лице,
правим тази размяна с
помощта на временна
променлива



Метод на прекия избор (пряка селекция)

$n-1$ пъти намираме индекса на най-малкия елемент от неподредената част на масива

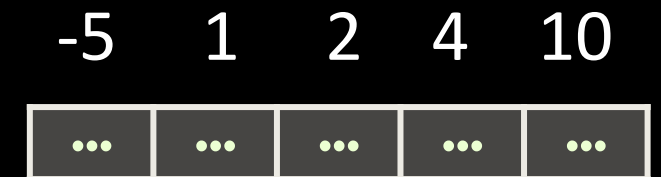
Масив от 5 елемента



■ Сложност $O(n^2)$

На всяко завъртане на цикъла поставяме най-малкия елемент на мястото му в подредената част на масива

Подреден масив



```
using System;

public class Program
{
    public static void Main()
    {
        int[] arr = new int[] { 2, 4, -5, 1, 10 };
        for (int i = 0; i < arr.Length; i++) {
            int k = i;
            for (int j = i + 1; j < arr.Length; j++) {
                if (arr[j] < arr[k])
                    k = j;
            }
            int swapVar = arr[i];
            arr[i] = arr[k];
            arr[k] = swapVar;
        }

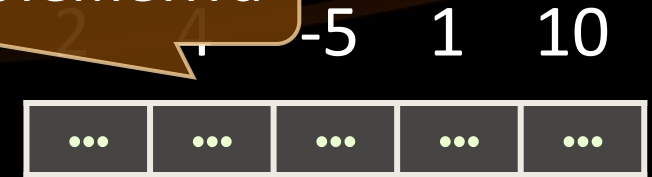
        Console.WriteLine(string.Join(" ", arr));
    }
}
```

Сортиране

$n-1$ пъти поставяме текущия елемент на мястото му в подредената част на масива

```
using System;
public class Program
{
    public static void Main()
    {
        int [] arr = { 2, 4, -5, 1, 10 };
        for (int i = 0; i < arr.Length; i++)
        {
            int swapVar = arr[i];
            int index = i;
            while (index > 0 && arr[index-1] >= swapVar)
            {
                arr[index] = arr[index-1];
                index--;
            }
            arr[index] = swapVar;
        }
        Console.WriteLine(string.Join(" ", arr));
    }
}
```

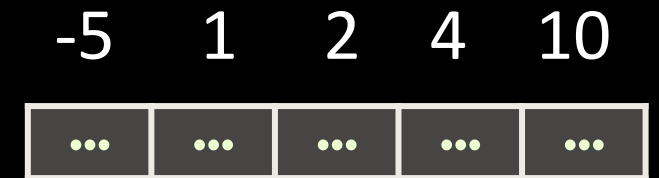
Масив от 5 елемента



■ Сложност $O(n^2)$

На всяко завъртане на цикъла ако текущия елемент е по-голям от дадения, проверяваме по-предния

Подреден масив



Задача: Сортиране на едномерен масив

- Напишете програма, която сортира елементите на масив от цели числа във възходящ ред
 - Масивът се въвежда от клавиатурата на един ред
 - Изходът е на един ред на екрана

{ 1 , -2, 7, -3, -4, 10, 2 } → { -4, -3, -2, 2, 1, 7, 10 }

Тествайте в Judge: <https://judge.softuni.bg/Contests/2655>

Решение: Сортиране на едномерен масив

- Ще приложим трети метод – **метода на вмъкването**
- Въвеждаме елементите на масива
- $n-1$ пъти **сравняваме съседните** елементи и **при необходимост** им **разменяме местата**

```
for (int i = 0; i < arr.Length - 1; i++)  
{  
    for (int j = 0; j < arr.Length - 1; j++)  
    {  
        if (arr[j] > arr[j + 1])  
        {  
            int swapVar = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] = swapVar;  
        }  
    }  
}
```

Тествайте в Judge: <https://judge.softuni.bg/Contests/2655>

C# Code – Как работи?

n-1 пъти **сравняваме**
съседните елементи

Ако **не са подредени**
правилно, то им
разменяме местата

```
for (int i = 0; i < arr.Length - 1; i++)  
{  
    for (int j = 0; j < arr.Length - 1; j++)  
    {  
        if (arr[j] > arr[j + 1])  
        {  
            int swapVar = arr[j];  
            arr[j] = arr[j + 1];  
            arr[j + 1] = swapVar;  
        }  
    }  
}
```

Какво научихме този час?

- Познати са различни методи за сортиране на масиви:
Метод на **мехурчето**, **пряка размяна**, **пряка селекция**, **бърза сортировка** и много други
- Сортирането се характеризира със **сложност** на алгоритъма и разход на **ресурс** (памет)
- Реалзацията му зависи от структурата от данни, която се ползва,
- При различен обем от данни, **различните** алгоритми са с **променливо** бързодействие и разход на ресурс



2 4 -5 1 10

... ..

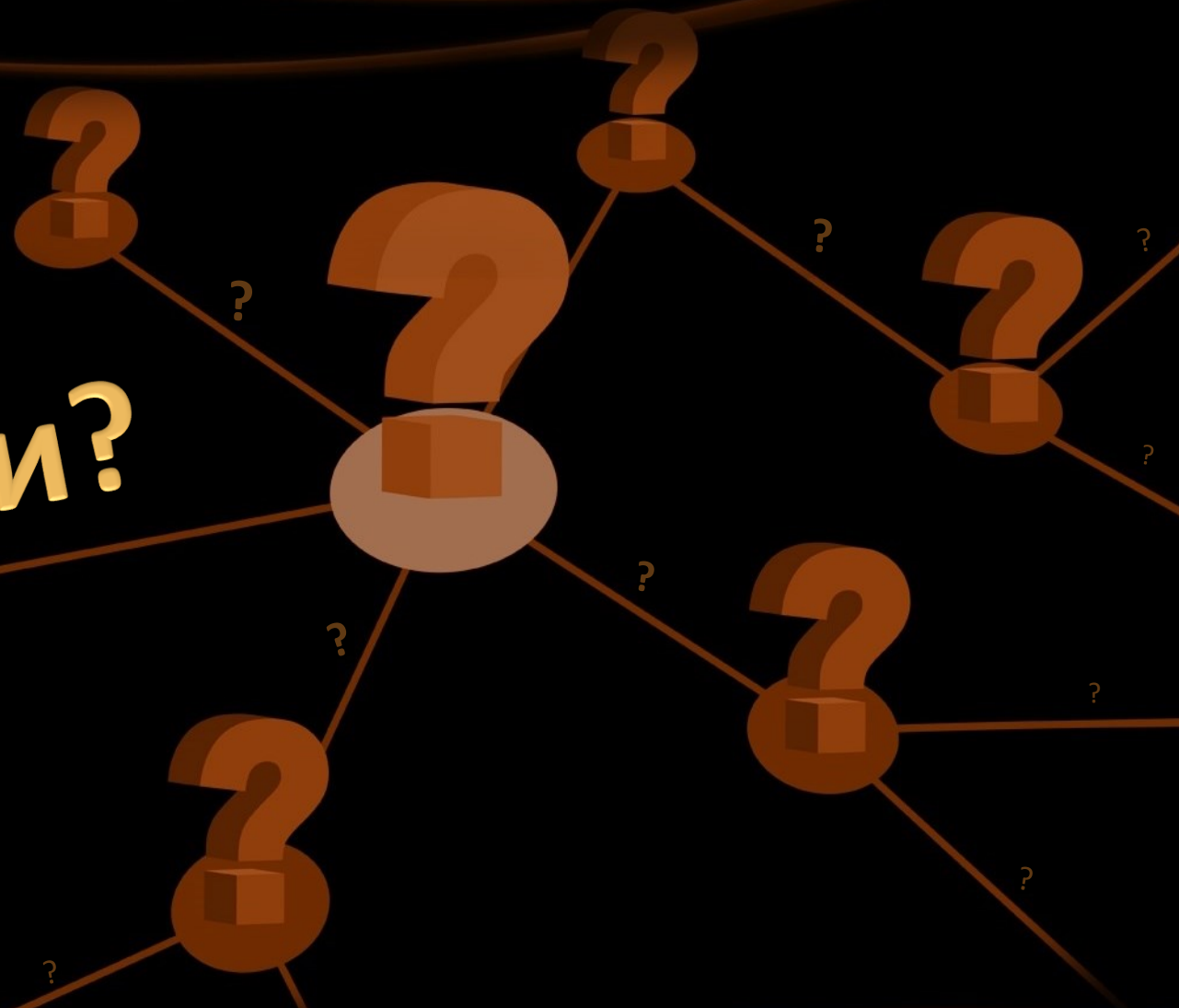
-5 1 2 4 10

... ..

Масиви



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

