

# Алгоритми

Емилиян Кадийски

<https://digitalrazgrad.org>  
<https://digitaltargovishte.org>

# Планът за днес

1. Какво е алгоритъм?
2. Алгоритми за сортиране на масив?
3. Сложност на алгоритми

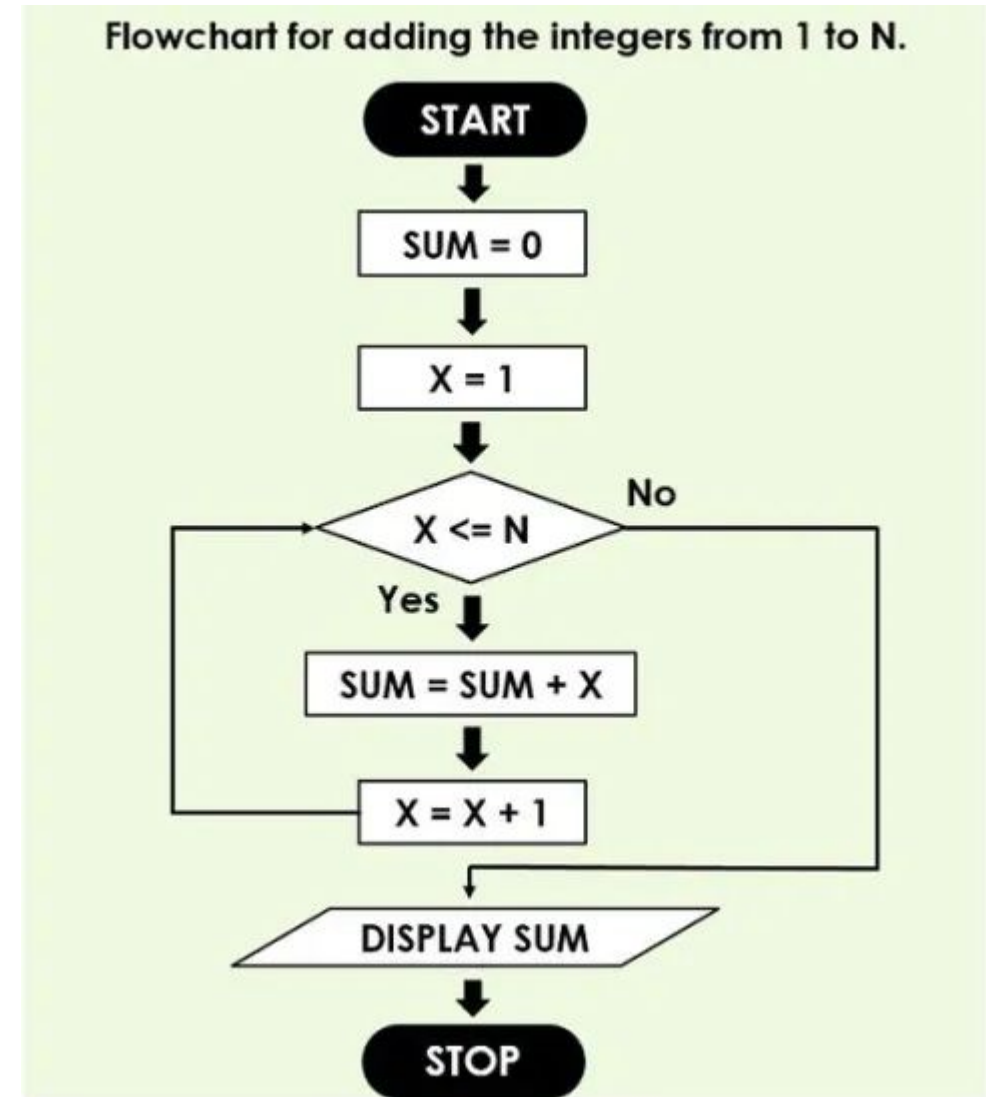


**АЛГОРИТЪМ**

?

# Какво е алгоритъм?

Описва сложно действие чрез редица от елементарни (достатъчно прости) действия, които изпълняващият може да извърши в последователни стъпки без допълнителни обяснения. Обикновено изпълнението на алгоритъма включва изчисление или обработка на данни.



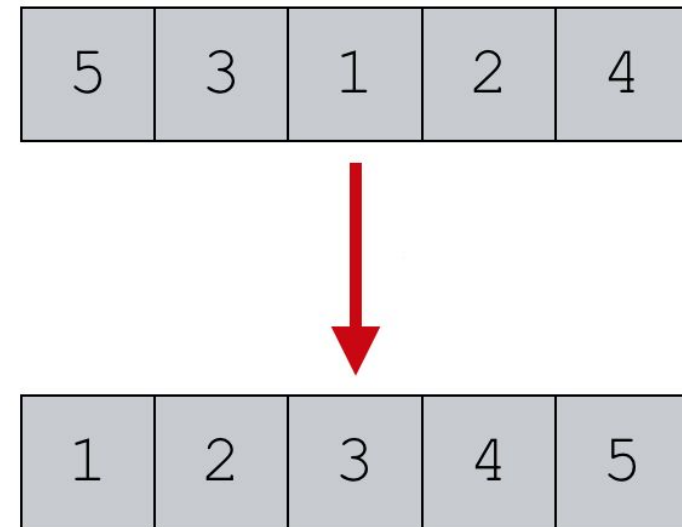
A solid dark blue circle is centered on the left side of the image. Inside the circle, the text "СОРТИРАНЕ НА МАСИВ" is written in white, bold, uppercase letters, arranged in three lines.

**СОРТИРАНЕ  
НА  
МАСИВ**

# Какво значи да сортираме масив?

Да сортираме един масив, означава да подредим елементите му според някакъв критерий.

Например имаме масив от оценки на студенти и трябва да ги подредим по нарастване - от най-ниските към най-високите.



# Алгоритъм на мехурчето

Логика:

- Елементите се сравняват се два по два
- Ако са в грешен ред, се разменят
- Трябва да се направят  $n*n$  итерации, за да сме сигурни, се всички елементи са в правилния ред за  $n$  = дължината на масива
- <https://www.youtube.com/watch?v=nmhjrl-aW5o>
- <https://www.youtube.com/watch?v=lyZQPjUT5B4&t=27s>

6 5 3 1 8 7 2 4

# Алгоритъм на мехурчето

```
int temp = 0;

for (int i = 0; i < array.length; i++) {
    for (int j = 1; j < (array.length - i); j++) {
        if (array[j - 1] > array[j]) {
            // swap the elements!
            temp = array[j - 1];
            array[j - 1] = array[j];
            array[j] = temp;
        }
    }
}

System.out.println(Arrays.toString(array));
```



# Insertion sort

Логика:

- Елементите се взимат един по един
- Всеки елемент се сравнява с тези преди него, докато му се намери правилната позиция
- След това се вмества и се продължава със следващия
- <https://www.youtube.com/watch?v=OGzPmgsl-pQ>
- <https://www.youtube.com/watch?v=ROaIU379I3U>

6 5 3 1 8 7 2 4

# Insertion sort

```
for (int i = 1; i < array.length; i++) {  
    int j = i;  
    int temp;  
    while (j > 0 && array[j - 1] > array[j]) {  
        temp = array[j - 1];  
        array[j - 1] = array[j];  
        array[j] = temp;  
        j = j - 1;  
    }  
}  
System.out.println(Arrays.toString(array));
```

# Сортиране чрез пряката селекция

Логика:

- Един по един се намира най-малкият елемент от списъка
- Разменя се с текущия елемент
- <https://www.youtube.com/watch?v=xWBP4IzkoyM>
- <https://www.youtube.com/watch?v=Ns4TPTC8whw>

	8
	5
	2
	6
	9
	3
	1
	4
	0
	7


# Сортиране чрез пряката селекция

```
for (int j = 0; j < array.length; j++) {  
    int minIndex = j;  
    for (int i = j + 1; i < array.length; i++) {  
        if (array[i] < array[minIndex]) {  
            minIndex = i;  
        }  
    }  
    int temp;  
    if (minIndex != j) {  
        temp = array[minIndex];  
        array[minIndex] = array[j];  
        array[j] = temp;  
    }  
}  
System.out.println(Arrays.toString(array));
```

# Задача

Разгледаните сортировки подреждат кода ни във възходящ ред. Преправете някоя от тях, така че кодът да е подреден в низходящ ред.



A dark blue circle containing the text "СЛОЖНОСТ НА АЛГОРИТМИ" in white, bold, uppercase letters.

# СЛОЖНОСТ НА АЛГОРИТМИ

# Защо да учим сложност на алгоритми?



2.037 милиарда активни потребители дневно

18 големи дейта центъра

20 млрд. \$ инвестиция

[Facebook data centers](#)

# Сложност на алгоритми

как се изменя изискваното време или памет за изпълнение на алгоритъм, с изменяне на размера на входните данни (за  $n$  - милиони и милиарди)



# Видове сложности

Сложност	Означение	Описание
Константна	$O(1)$	не зависи от входните данни
Логаритмична	$O(\log(N))$	Има цикъл с * или \
Линейна	$O(N)$	Има 1 цикъл
Енлог	$O(N \cdot \log(N))$	Има 2 цикъла, единия е с * или \
Квадратична	$O(N^2)$	Има два вложени цикъла
Кубична	$O(N^3)$	Има три вложени цикъла
Експоненциална	$O(2^N)$	

# Нарастване на най-често използваните функции

Функция / $n$	1	2	10	100	1000
5	5	5	5	5	5
$\log n$	0	1	3.32	6.64	9.96
$n$	1	2	10	100	1000
$n \log n$	0	2	33.2	664	9996
$n^2$	1	4	100	$10^4$	$10^6$
$n^3$	1	8	1000	$10^6$	$10^9$
$2^n$	2	4	1024	$10^{30}$	$10^{300}$
$n!$	1	2	3628800	$10^{157}$	$10^{2567}$
$n^n$	1	4	$10^{10}$	$10^{200}$	$10^{3000}$

# Задачи за упражнение

Примерни задачи

Самостоятелни задачи

# Trainings @ Digital Razgrad & Digital Targovishte

- Digital Razgrad
  - <https://digitalrazgrad.org>
  - <https://facebook.com/digitalrazgrad.org>
  - [digitalrazgrad.slack.com](https://digitalrazgrad.slack.com)
- Digital Targovishte
  - <https://digitaltargovishte.org>
  - <https://facebook.com/digitaltargovishte.org>
  - [digitaltargovishte.slack.com](https://digitaltargovishte.slack.com)

