

University of Ljubljana  
Faculty *of Computer and  
Information Science*



# Uvod v računalništvo

Vaje

29.11 -  
3.12.2021



## Atributi algoritmov

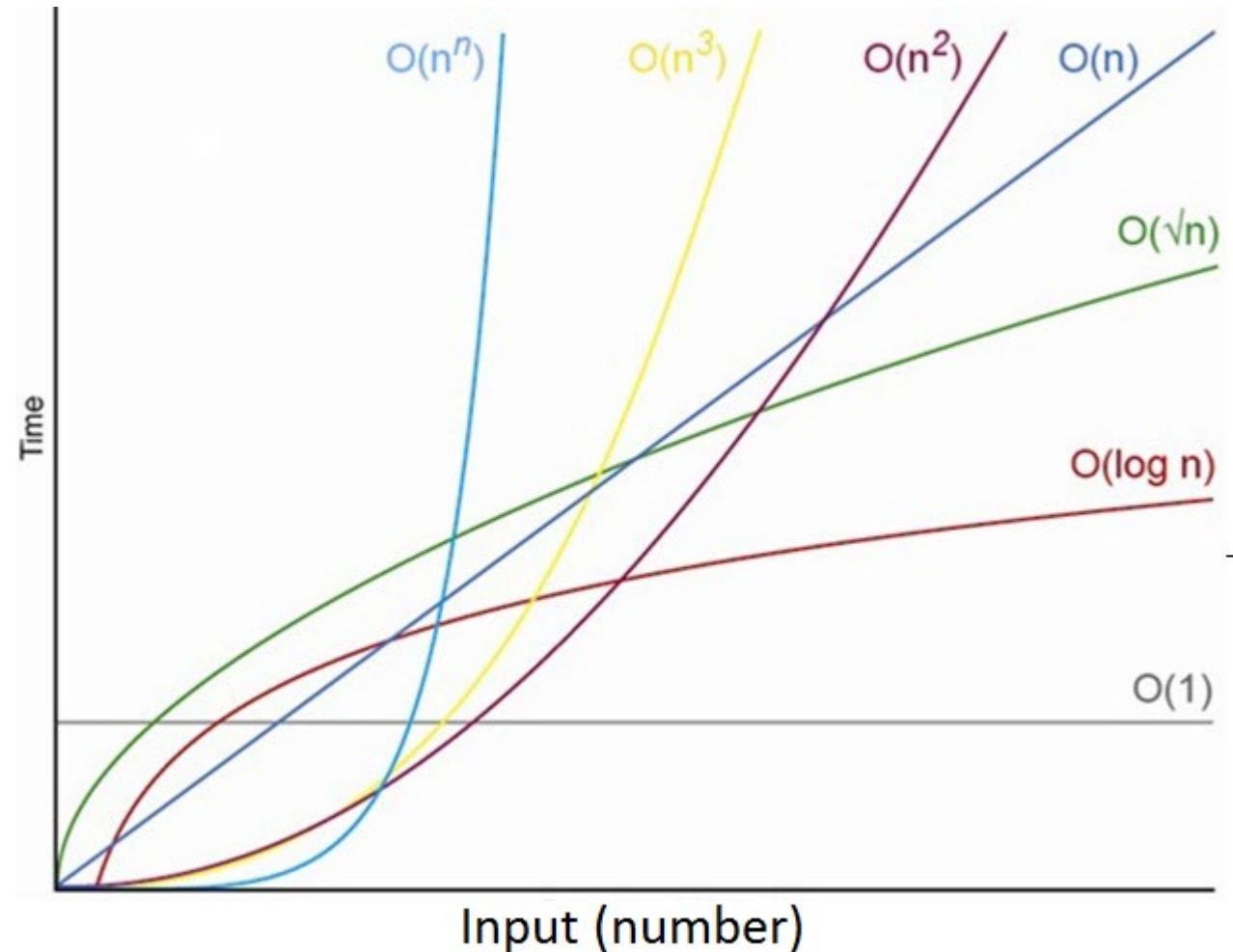
*Kaj je dober algoritem?*

- **Pravilnost** – Algoritem mora dati pravilen rezultat, pravilnemu problemu in se nato ustaviti.
- **Razumljivost** - Algoritem mora biti enostaven za razumevanje, da bo vzdrževanje čim lažje.
- **Eleganca** - kompakten, jasen in učinkovit algoritem.
- **Učinkovitost** - optimalna izraba virov.
  - časovna učinkovitost/potratnost - globalna indikacija količine "dela", ki jo zahteva algoritem
  - prostorska učinkovitost/potratnost - količino informacij, ki jih mora algoritem shraniti v pomnilnik računalnika



## Časovna učinkovitost

Big O	Naziv
$O(1)$	Constant
$O(\log n)$	Logarithmic
$O(n)$	Linear
$O(n \log n)$	Linearithmic
$O(n^2)$	Quadratic
$O(n^3)$	Cubic
$O(2^n)$	Exponential
$O(n!)$	Factorial





## Časovna učinkovitost - primeri

### $O(1)$

```
function sum(a,b){  
    return a+b;  
}
```

### $O(n)$

```
int i, y;  
for ( i = 0; i < n; ++i ) {  
    y = i + 1;  
}
```

### $O(\log n)$

```
x = n  
while ( x > 0 ) {  
    x = x / 2  
}
```

### $O(n^2)$

```
x = n;  
while ( x > 0 ) {  
    y = x;  
    while ( y > 0 ) {  
        y = y - 1;  
    }  
    x = x - 1;  
}
```

### $O(2^n)$

```
function fibonacci(num) {  
    if(num < 2) {  
        return num;  
    }  
    else {  
        return fibonacci(num-1) + fibonacci(num - 2);  
    }  
}
```

### $O(n!)$

```
function factorial(n){  
    int i = n;  
    for (i = 0; i < n; i++) {  
        print(n);  
        factorial(n-1);  
    }  
}
```



## Naloga 8.1

V kateri razred zahtevnosti sodi algoritem v primeru, da:

- a) Za osnovno enoto dela izberemo odštevanje?
- b) Za osnovno enoto dela izberemo celoštevilsko deljenje?

Utemelji odgovore!

Preberi vrednost  $n$

Nastavi  $c$  na  $0$

Nastavi  $a$  na vrednost  $n$

Dokler  $a \geq 1$ , ponavljaaj

$c = c + 1$

$b = n$

Dokler  $b > 0$ , ponavljaaj

$c = c + 1$

$b = b / 2$

$a = a - 1$

Ustavi se.



## Naloga 8.1

V kateri razred zahtevnosti sodi algoritem v primeru, da:

- a) Za osnovno enoto dela izberemo odštevanje?
- b) Za osnovno enoto dela izberemo celoštevilsko deljenje?

Utemelji odgovore!

a)  $O(n)$

b)  $O(n \cdot \log n)$

Preberi vrednost  $n$

Nastavi  $c$  na  $0$

Nastavi  $a$  na vrednost  $n$

Dokler  $a \geq 1$ , ponavljaaj

$c = c + 1$

$b = n$

Dokler  $b > 0$ , ponavljaaj

$c = c + 1$

$b = b / 2$

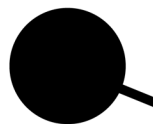
$a = a - 1$

Ustavi se.



## Naloga 8.2

Algoritma A in B rešujeta isti problem. Pri vrodu velikosti  $n$  algoritem A izvrši  $0,003n^2$  operacij, algoritem B pa  $243n$  operacij. Poišči vrednost  $n$ , nad katero algoritem B postane bolj učinkovit kot algoritem A.





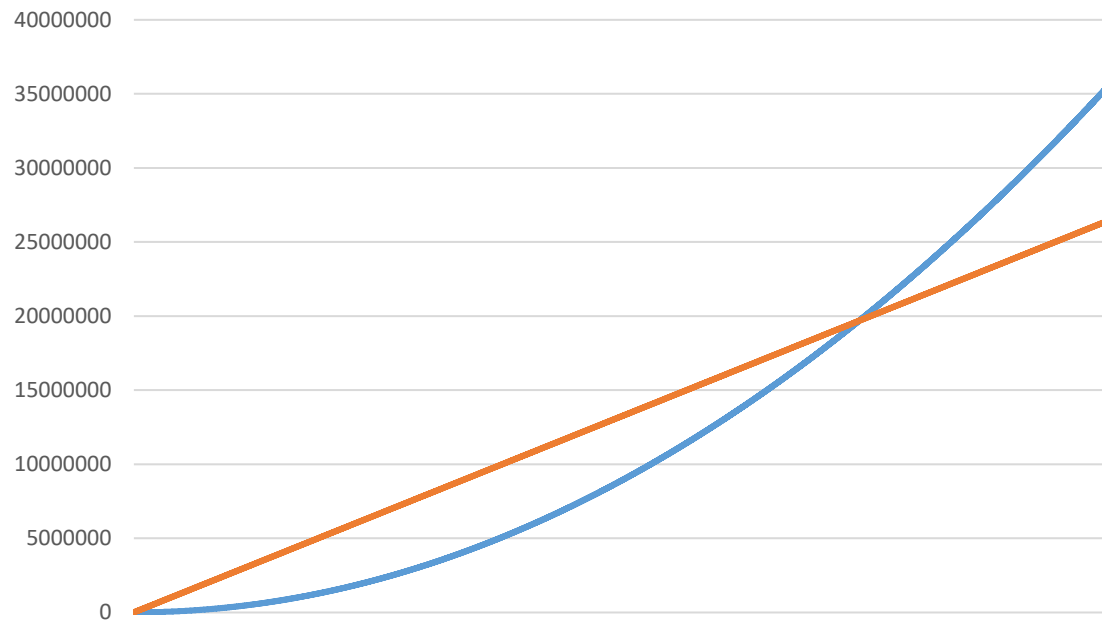
## Naloga 8.2

$$0.003 n^2 = 243 n$$

$$0.003 n = 243$$

$$n = 243 / 0.003$$

$$n = 81000$$



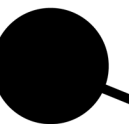
=> Pri vrednostih  $n$ -ja večjih od 81000 bo algoritem B učinkovitejši od algoritma A.





## Naloga 8.3

Algoritem, katerega časovna zahtevnost je reda velikosti  $O(n)$ , se pri vrednosti  $n = 100$  na določenem računalniku izvaja 10 sekund. Kako dolgo naj bi se izvajal pri vrednosti  $n = 500$ ?





## Naloga 8.3

Algoritem, katerega časovna zahtevnost je reda velikosti  $O(n)$ , se pri vrednosti  $n = 100$  na določenem računalniku izvaja 10 sekund. Kako dolgo naj bi se izvajal pri vrednosti  $n = 500$ ?

100

10

500

x

---

$$x = (10 \cdot 500) / 100$$

$$x = 50 \text{ s}$$



## Naloga 8.4

Podana je nekoliko posplošena različica algoritma za zaporedno iskanje oseb v telefonskem imeniku.

- S pomočjo zgornjega algoritma poišči telefonsko številko Dumbledora v naslednjem imeniku.
- V kateri razred časovne zahtevnosti uvrščamo ta algoritem? Utemelji.

Preberi vrednosti za  $IME$ ,  $n$ ,  $N_1$ ,  $N_2$ , ...,  $N_n$  in  $T_1$ ,  $T_2$ , ...,  $T_n$ .

Nastavi vrednost  $i$  na 1 in vrednost  $Najden$  na NE.

Dokler  $((Najden = NE) \wedge (i \leq n))$ , ponavlajaj

    Če se  $IME$  ujema z  $i$ -tim imenom v imeniku,  $N_i$

        Izpiši telefonsko številko te osebe,  $T_i$ .

        Nastavi vrednost  $Najden$  na DA.

    sicer (t.j.  $IME$  se ne ujema z  $N_i$ )

$i$  povečaj za 1.

Če  $(Najden = NE)$

    Izpiši sporočilo "Oprostite, tega naročnika ni v imeniku."

Ustavi se.

Oseba	Telefonska številka
Harry	03 3940510
Ron	04 9753020
Hermione	02 7772892
Hagrid	08 8829374
Dumbledore	04 8264950
Malfoy	01 6624584



## Naloga 8.4

`i=1`

`Najden = NE`

`Dokler (Najden == NE) IN (i<=6) ponavlaj`

`Dumblerdore != N1`

`i=i+1`

`Dumblerdore != N2`

`i=i+1`

`***`

`Dumblerdore = N5`

`Telefonska številka je 04 8264950`

`Najden = DA`

Oseba	Telefonska številka
Harry	03 3940510
Ron	04 9753020
Hermione	02 7772892
Hagrid	08 8829374
Dumblerdore	04 8264950
Malfoy	01 6624584

Vstavitev

$O(n)$



## Naloga 8.5

Spodaj je podan algoritem za mehurčno urejanje (angl. bubble sort).

Preberi vrednost  $n$  za velikost seznama.

Preberi vseh  $n$  elementov seznama.

Postavi oznako  $N$  za neurejeni del seznama na konec seznama.

Dokler velja, da ima neurejeni del seznama vsaj dva elementa, ponavljaj

Postavi oznako  $T$  za trenutni element na drugi element seznama.

Dokler oznaka  $T$  ne preskoči oznake  $N$ , ponavljaj

Če je element na položaju  $T$  manjši od elementa na njegovi levi  
zamenjaj ta dva elementa

Premakni oznako  $T$  eno mesto v desno.

Premakni oznako  $N$  eno mesto v levo.

Ustavi se.

a) Zapiši seznam: 35 83 44 12 76 91 55 7 27 31, po vsakem obhodu zunanje zanke. V kateri razred časovne zahtevnosti uvrščamo ta algoritem? Utemelji.

b) Recimo, da izvedemo algoritem za urejanje z izbiranjem in algoritem za mehurčno urejanje na seznamu, ki je že pravilno urejen po velikosti. Pri katerem izmed obeh algoritmov imamo manj zamenjav elementov? Utemelji?



## Naloga 8.5

a)

35	<u>83</u>	44	12	76	<u>91</u>	55	7	27	31
----	-----------	----	----	----	-----------	----	---	----	----

po 1 obhodu:

35	44	<u>12</u>	76	<b>83</b>	55	7	27	31	<b>91</b>
----	----	-----------	----	-----------	----	---	----	----	-----------

po 2 obhodih:

35	<b>12</b>	44	<u>76</u>	55	7	27	31	<b>83</b>	91
----	-----------	----	-----------	----	---	----	----	-----------	----

po 3 obhodov:

35	12	44	<u>55</u>	7	27	31	<b>76</b>	83	91
----	----	----	-----------	---	----	----	-----------	----	----

\*\*\*

b)

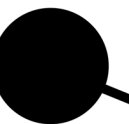
**Urejanje z izbiranjem:** naredi po eno zamenjavo za vsak položaj na seznam in ne upošteva morebitni že obstoječe urejenosti.

**Mehurčno urejanje:** zamenja le elemente, ki niso pravilno urejeni. Če je seznam že urejen, ne bo zamenjave.



## Dodatna naloga 1

Algoritem, katerega časovna zahtevnost je reda velikosti  $O(n^2)$ , se pri vrednosti  $n = 100$  na določenem računalniku izvaja 10 sekund. Kako dolgo naj bi se izvajal pri vrednosti  $n = 500$ ?





## Dodatna naloga 1

Algoritem, katerega časovna zahtevnost je reda velikosti  $O(n^2)$ , se pri vrednosti  $n = 100$  na določenem računalniku izvaja 10 sekund. Kako dolgo naj bi se izvajal pri vrednosti  $n = 500$ ?

$$100^2$$

$$10$$

$$500^2$$

$$x$$

---

$$x = \frac{10 * 250000}{10000}$$

$$x = 250 \text{ s}$$





## Dodatna naloga 2

V kateri razred zahtevnosti sodi algoritem v primeru, da:

- a) Za osnovno enoto dela izberemo seštevanje?  
Utemelji odgovore!

Preberi vrednost  $n$

Nastavi  $c$  na  $0$

Nastavi  $a$  na vrednost  $n$

Dokler  $a \geq 1$ , ponavlaj

$c = c + 1$

$b = n$

Dokler  $b > 0$ , ponavlaj

$c = c + 1$

$b = b / 2$

$a = a - 1$

Ustavi se.



## Dodatna naloga 2

V kateri razred zahtevnosti sodi algoritem v primeru, da:

a) Za osnovno enoto dela izberemo seštevanje?  
Utemelji odgovore!

a)  $O(n \cdot \log n)$

Preberi vrednost  $n$

Nastavi  $c$  na  $0$

Nastavi  $a$  na vrednost  $n$

Dokler  $a \geq 1$ , ponavlaj

$c = c + 1$

$b = n$

Dokler  $b > 0$ , ponavlaj

$c = c + 1$

$b = b / 2$

$a = a - 1$

Ustavi se.



## Dodatna naloga 3

V kateri razred zahtevnosti sodi algoritem?

Utemelji odgovore!

Preberi vrednost  $n$

$i = n$

Dokler  $i > 0$ , ponavljaj

$i = i / 2$

$j = 1$

Dokler  $j < n$ , ponavljaj

$j = j * 2$

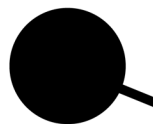
$k = 0$

Dokler  $k < n$ , ponavljaj

$k = k + 4$

(konstantno število operacij)

Ustavi se.





## Dodatna naloga 3

V kateri razred zahtevnosti sodi algoritem?

Utemelji odgovore!

Preberi vrednost  $n$

$i = n$

Dokler  $\frac{i > 1, \text{ ponavljaj } \left\{ \begin{array}{l} i = i / 2 \\ j = 1 \end{array} \right\}}{j = 1} O(\log n)$

Dokler  $\frac{j < n, \text{ ponavljaj } \left\{ \begin{array}{l} j = j * 2 \\ k = 0 \end{array} \right\}}{k = 0} O(\log n)$

Dokler  $\frac{k < n, \text{ ponavljaj } \left\{ \begin{array}{l} k = k + 4 \end{array} \right\}}{k = k + 4} O(n)$

(konstantno število operacij)

Ustavi se.

$$\left. \begin{array}{l} O(\log n) \\ O(\log n) \\ O(n) \end{array} \right\} O(n * \log n * \log n) = O(n \log^2 n)$$



# Hvala za pozornost!

Petar.Kochovski@fri.uni-lj.si