

# Algoritmi in podatkovne strukture 1

Visokošolski strokovni študij Računalništvo in informatika

Algoritmi  
in problemi



# Algoritmi

- Izvor izraza
  - al-Khwārizmī → ... → algoritmi

*Sem*

*Muhammad ibn Mūsā al-Khwārizmī  
Perzijski matematik iz 9. stoletja.*

*Opisal sem algoritme za seštevanje,  
množenje, deljenje, kvadratni koren,  
izračun decimalk  $\pi$ , itd. števil  
v Arabskem (Indijskem) zapisu.*



# Algoritmi

- Primer algoritma: seštevanje celih števil

## Seštevanje dveh števil

- števili napišemo eno pod drugo
- začnemo pri enicah
- seštevamo po dve števki
- upoštevamo prenos
- se pomikamo po mestih proti levi

$$\begin{array}{r} 1357 \\ + 2846 \\ \hline 1110 \\ \hline 4203 \end{array}$$

*Zakaj ta algoritem  
sploh deluje?*



# Algoritmi

- Poskus algoritma: seštevanje celih števil
  - Evropa v času, ko je živel al-Khwārizmī, uporablja rimske številke!

ROMAN NUMERALS

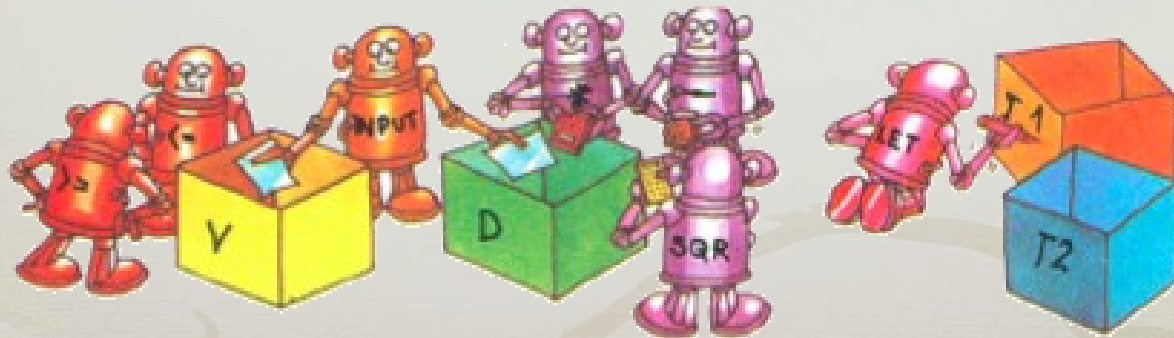
I	II	III	IV	V
1	2	3	4	5
VI	VII	VIII	IX	X
6	7	8	9	10
XI	XII	XIII	XIV	XV
11	12	13	14	15
XVI	XVII	XVIII	XIX	XX
16	17	18	19	20
XXV	XXX	XXXV	XL	XLV
25	30	35	40	45
L	LX	LXX	LXXX	XC
50	60	70	80	90
C	CXXV	CCL	D	M
100	125	250	500	1000

MCDLI  
+ DLXVII  
-----  
MMXVIII

M=1000, D=500, C=100,  
L=50, X=10, V=5, I=1

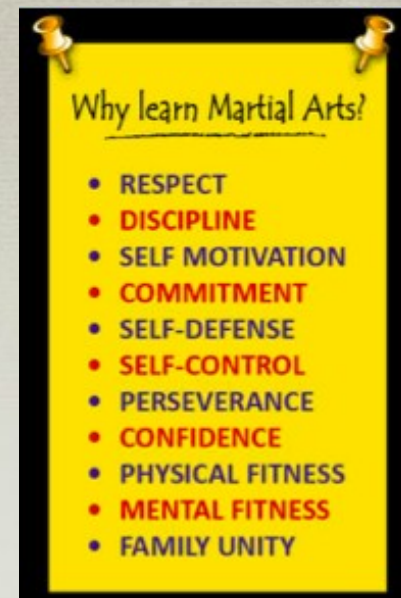
# Algoritmi

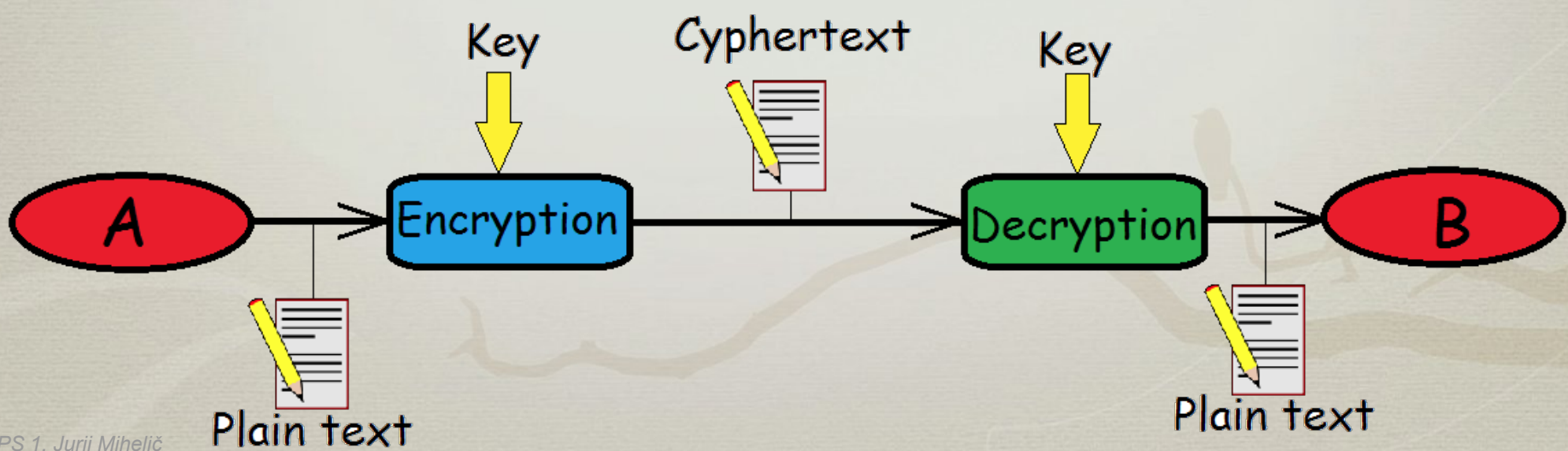
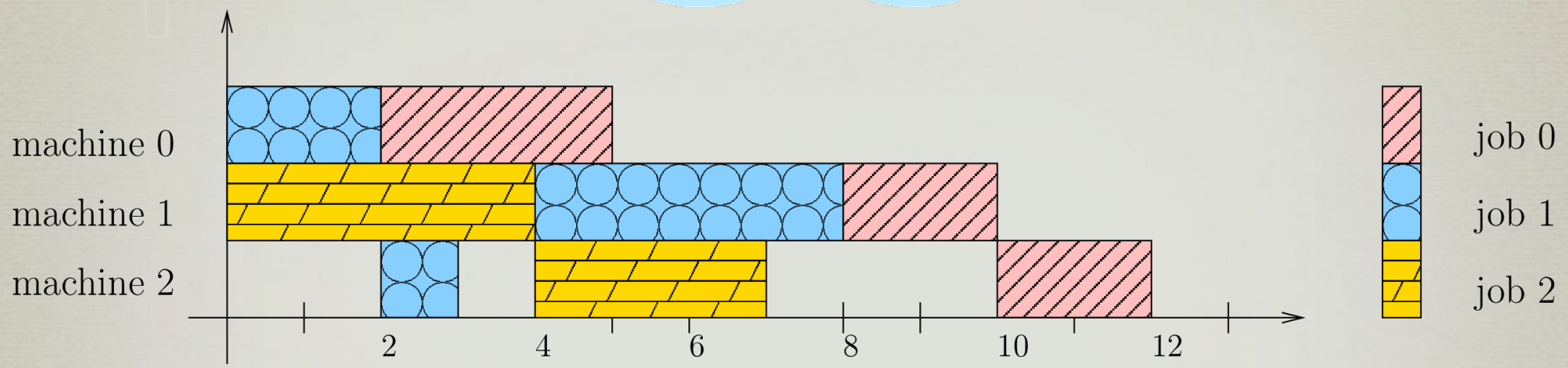
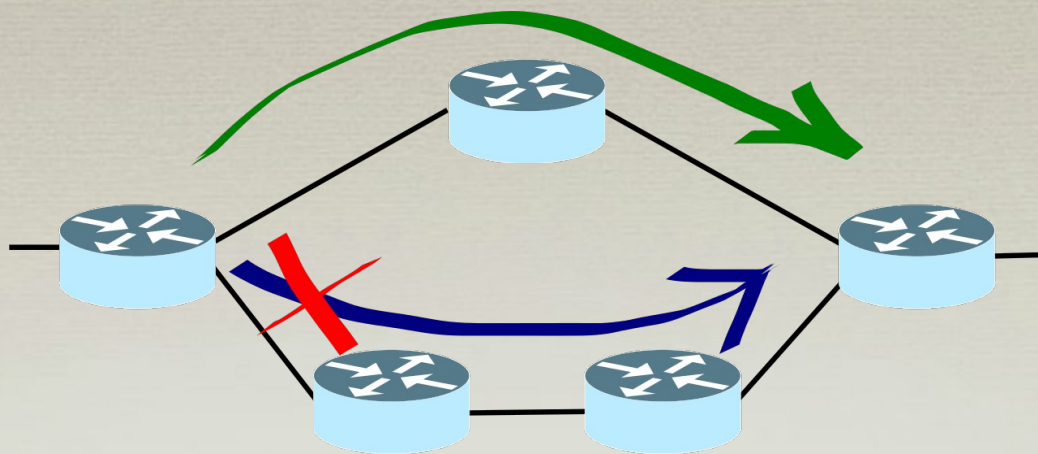
- Osnovna vprašanja
  - Kaj je algoritem?
  - Kaj je vhod algoritma?
  - Kaj je izhod algoritma?
  - Kaj je (računski) problem?
  - Kaj pomeni rešiti nek problem?
  - itd.

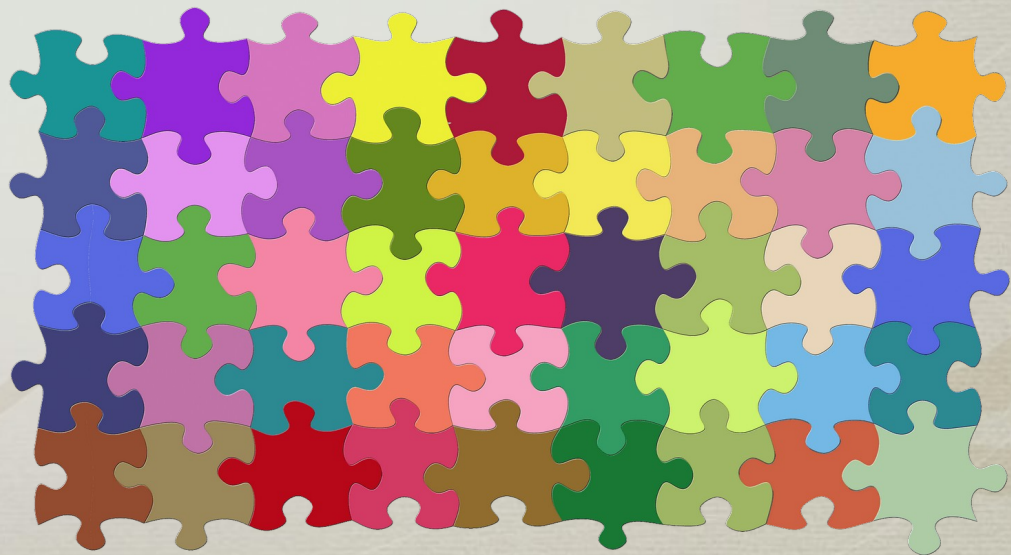
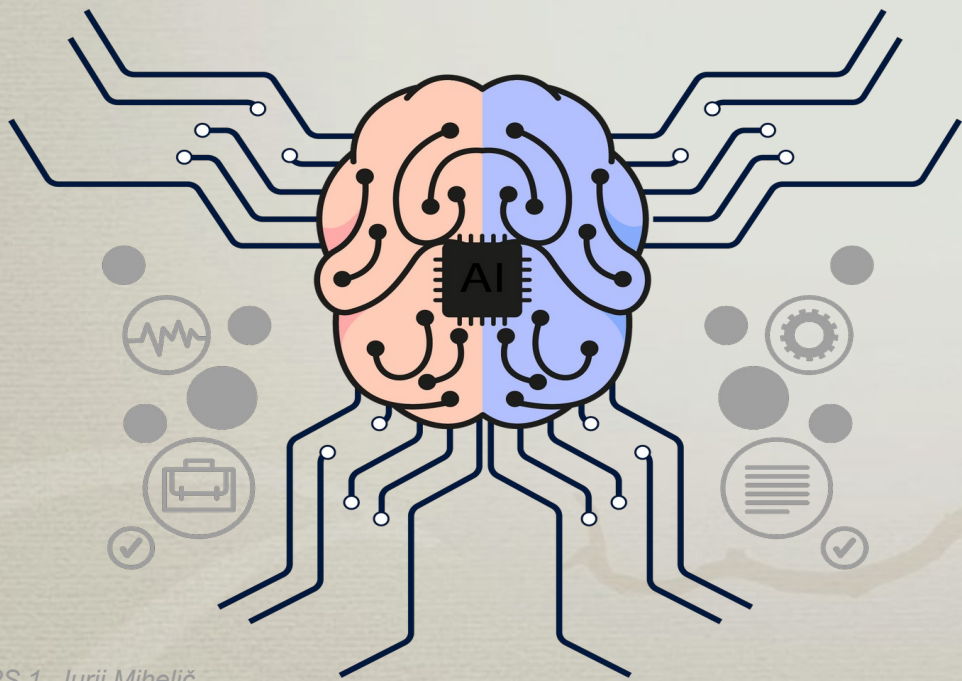
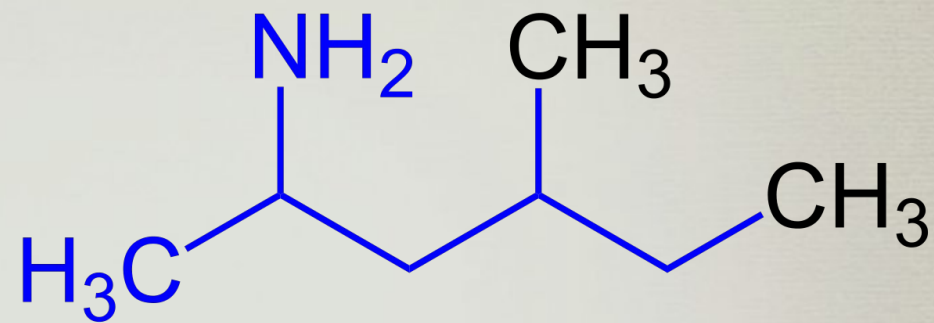
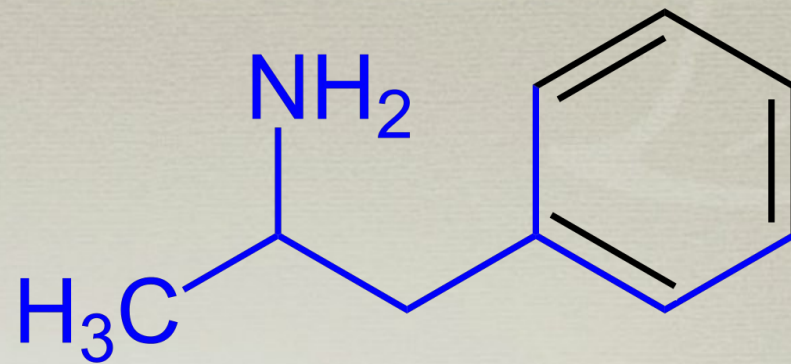
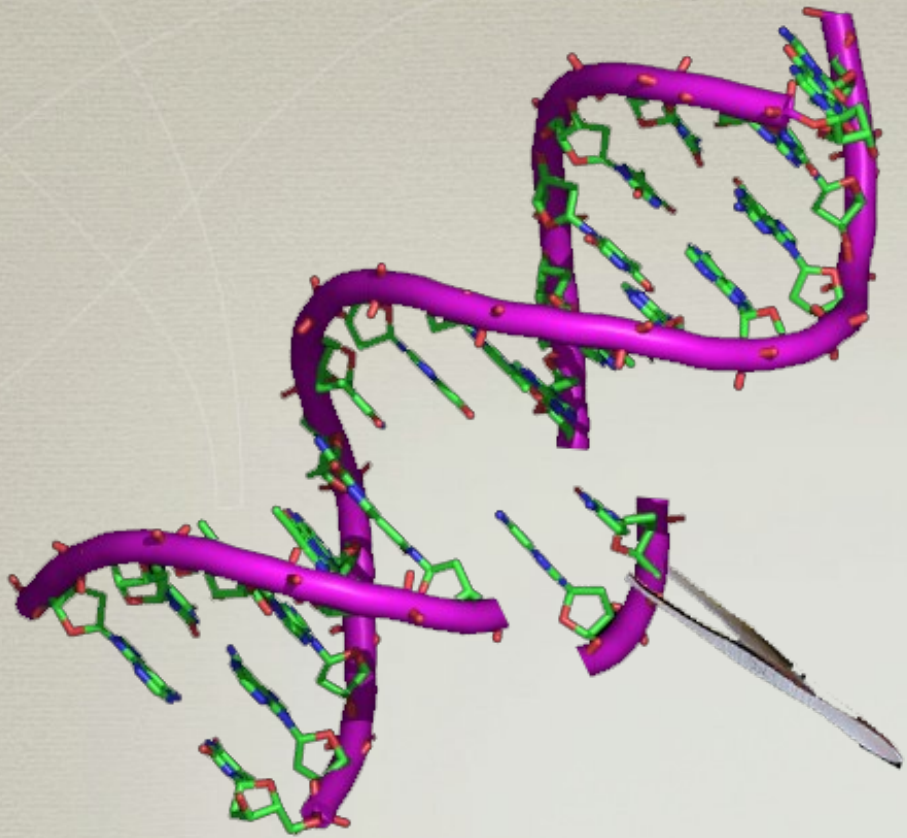


# Algoritmi

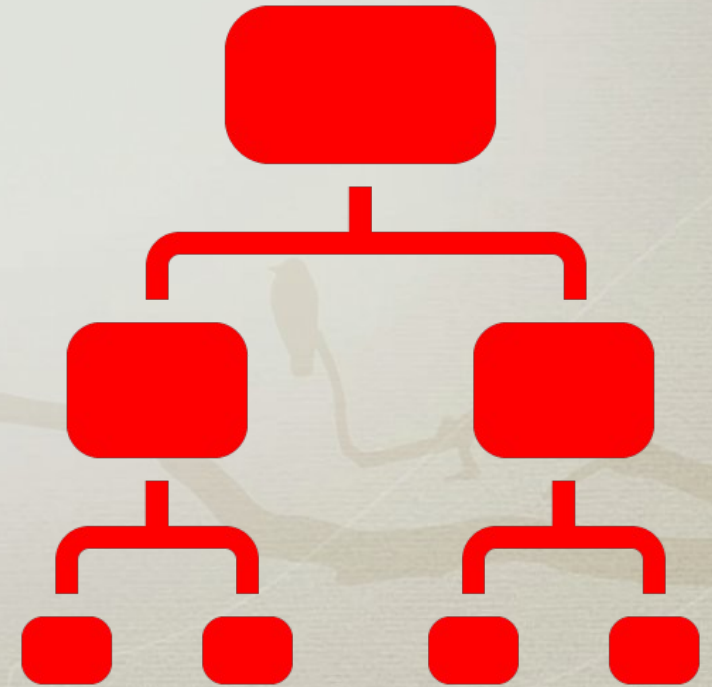
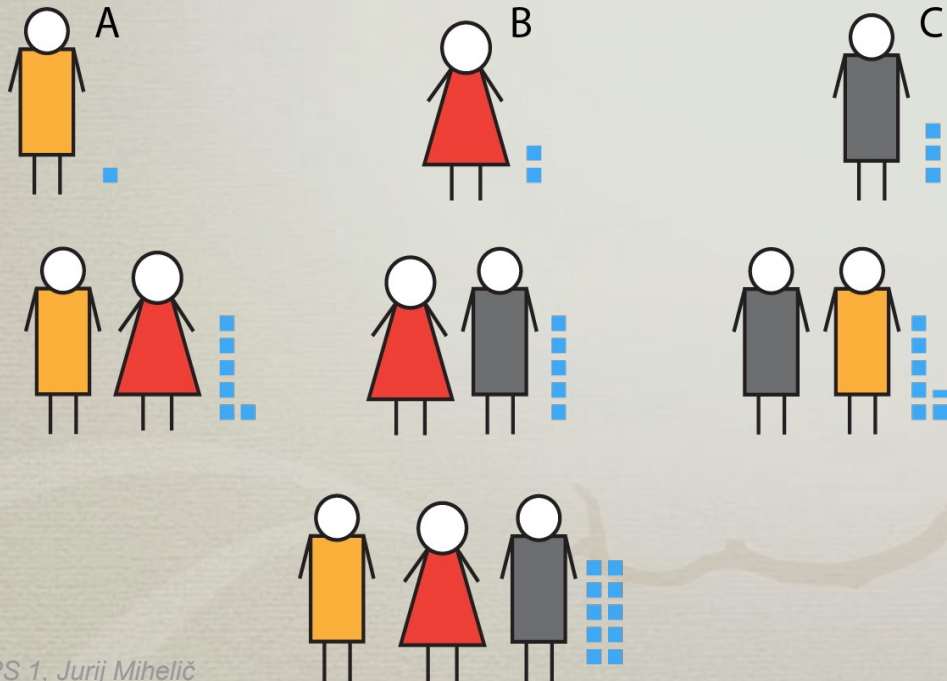
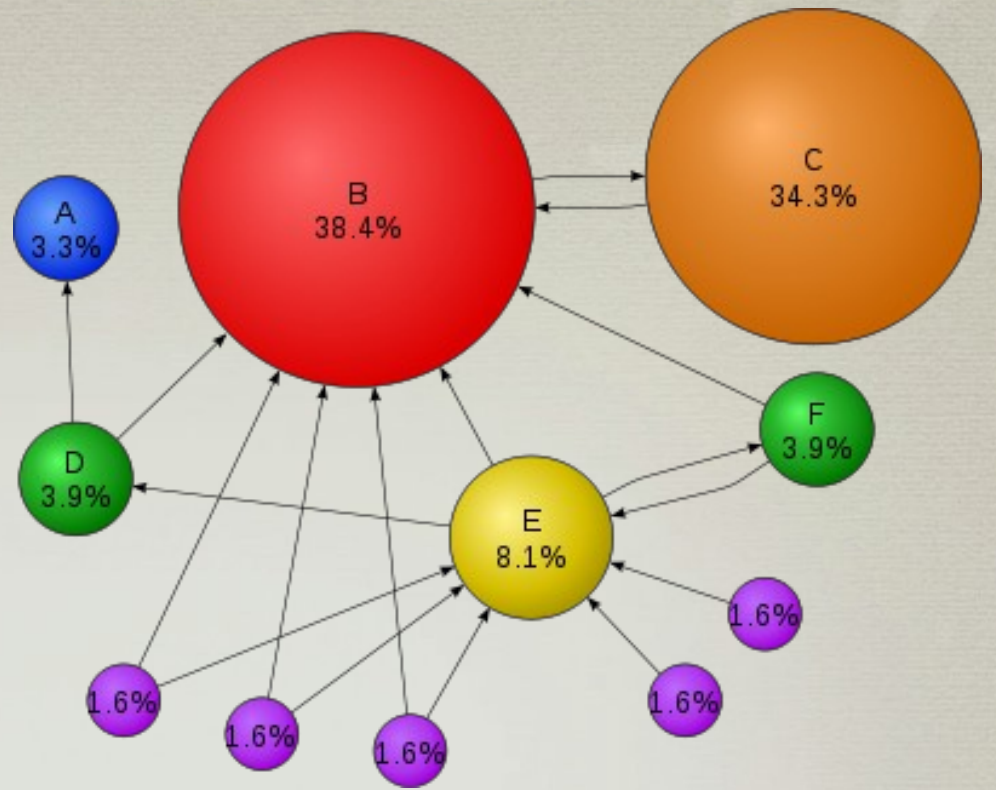
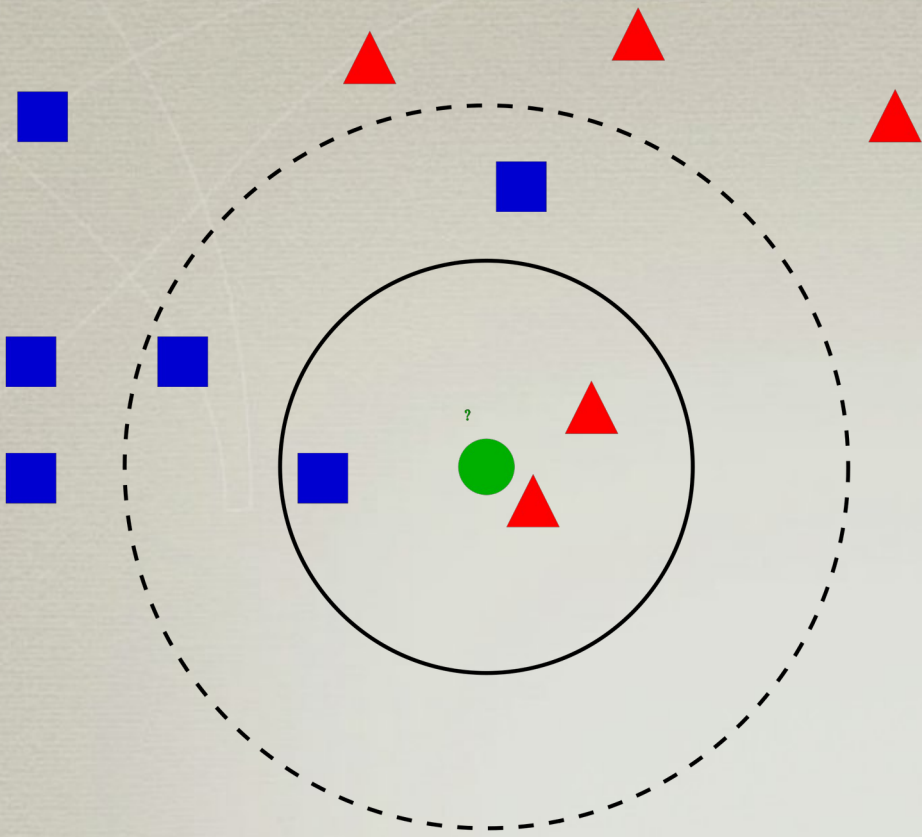
- Zakaj bi se ukvarjali z algoritmi?
  - algoritmi so **temelj računalništva**
    - se pojavljajo praktično povsod
  - algoritmi krojijo **življenje in svet**
    - rangiranje strani, prikazovanje reklam, priporočilni sistemi, avtonomna vožnja
  - Interdisciplinarnost področja
    - podatkovna znanost, omrežne vede, umetna inteligenca
  - razumevanje ozadja programskih knjižnic
    - `Arrays.sort(...)`, `sorted(...)`
  - radovednost, biti boljši programer











# Algoritmi

Algoritem je  
**jasen**, **nedvoumen** in mehaničen postopek  
za reševanje *računskega problema*.

Neformalna, intuitivno definicija, ki zadošča za potrebe APS.

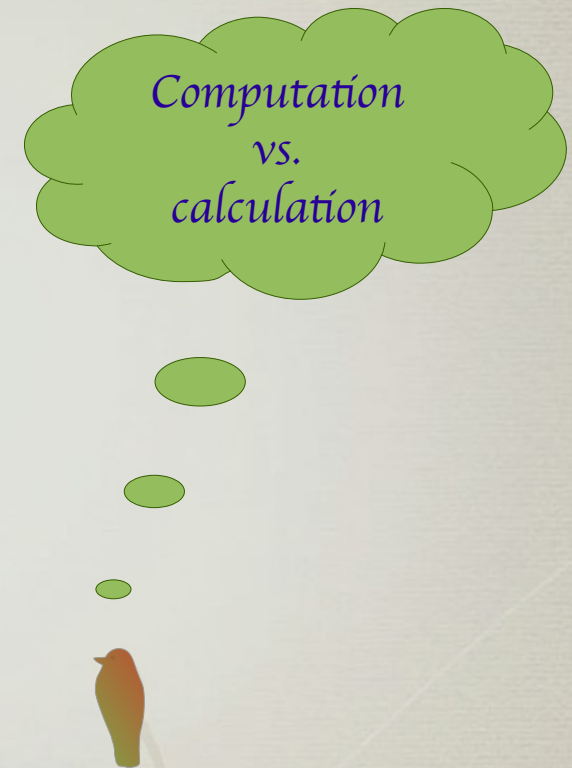


# Algoritmi

- Algoritem je
  - **jasen**
    - enostavno razumljiv, skladen z okoljem oz. znanjem
  - **nedvoumen**
    - razumljiv na samo en način, determinističen, enoličen
  - **mehaničen**
    - uporablja elementarne operacije, katerih izvajanje ne zahteva genialnosti
  - **postopek**
    - končen opis izvajanja, korak za korakom
  - za reševanje **računskega problema**

# Računski problemi

- Rešiti računski problem pomeni,
  - za **vse** možne **naloge** problema
  - znati poiskati ustrezno **rešitev**.
- Primer problema
  - seštevanje števil do 100
  - za dani celi števili  $x$  in  $y$ ,  
kjer  $0 \leq x, y \leq 100$
  - izračunaj njuno vsoto  $x + y$



Ali gre le za računanje (calculation) ali za ka več (computation)?

# Računski problemi

- Računski problem
  - splošno in natančno opisuje želeni odnos
  - med *nalogami* in
  - njihovimi *rešitvami*.

# *Računski problemi*

- Naloge in rešitve
  - naloga je konkreten primerek problema
  - rešitev se nanaša na neko nalogo

## **Primer:**

**problem:** seštevanje števil do 100

**naloga:**  $12 + 13$ , **rešitev:** 25

**naloga:**  $100 + 0$ , **rešitev:** 100

**ni naloga:**  $101 + 1$

PS: Koliko je možnih nalog in koliko rešitev?

# *Računski problemi*

- Algoritem torej rešuje računski problem
  - kadar zna za vsako nalogo problema
  - izračunati pravilno rešitev



# Računski problemi

- Vrste računskih problemov

- **iskalni**

- rešitev je objekt, ki zadošča danim omejitvam
      - Za dani števili izračunaj njuno vsoto.
      - Uredi podani seznam elementov.
      - Poišči pot iz predavalnice do menze.

- **odločitveni**

- rešitev je oblike da/ne
      - Za podana števila  $x$ ,  $y$ , in  $z$  preveri ali je  $x+y=z$ ?
      - Ali je podano zaporedje števil urejeno?
      - Ali obstaja pot iz predavalnice do menze krajša od 42 m?



# Računski problemi

- Vrste računskih problemov
  - **preštevalni**
    - rešitev je število objektov, ki zadoščajo danim omejitvam
      - Na koliko različnih načinov lahko dve naravni števili seštejemo, da dobimo 42?
      - Koliko je različnih razvrstitev elementov danega seznama?
  - **naštevalni**
    - podobno kot preštevalni, le da je rešitev množica objektov, ki zadoščajo danim omejitvam
  - **optimizacijski**
    - iskanje najboljše rešitve izmed vseh možnih
    - *kriterijski* – iščemo le vrednost rešitve
    - *konstrukcijski* – rešitev želimo konstruirati



# *Računski problemi*

- Formalna definicija računskega problema
  - množica vseh parov
    - naloga problema
    - in njena rešitev
- Formalna definicija algoritma
  - Turingov stroj, ..., programski jeziki
- Church-Turingova teza
  - S Turingovim strojem se da izračunati vse, kar se sploh da izračunati.

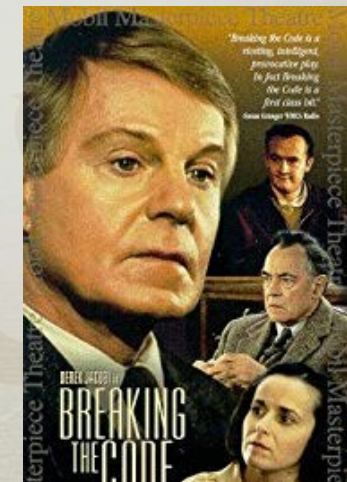


# Računski problemi



Alan Turing, 1912 - 1954

- Alan Turing
  - utemeljitelj algoritmike, izračunljivosti, ...
  - razbil nemško kodo Enigma
  - umetna inteligenca: Turingov test
  - filmi o Turingu
    - Imitation game, 2014; Codebreaker, 2011; Breaking the Code, 1996



# Snovanje algoritmov

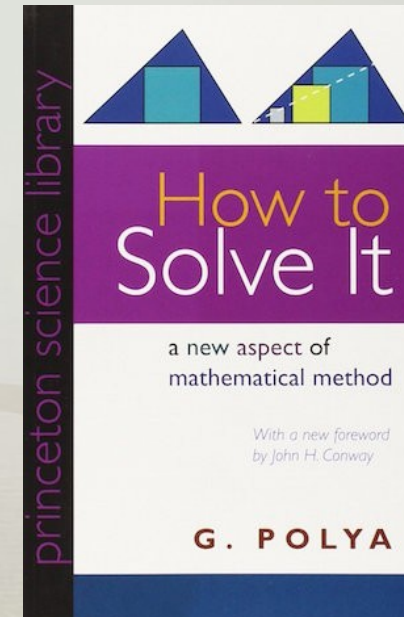
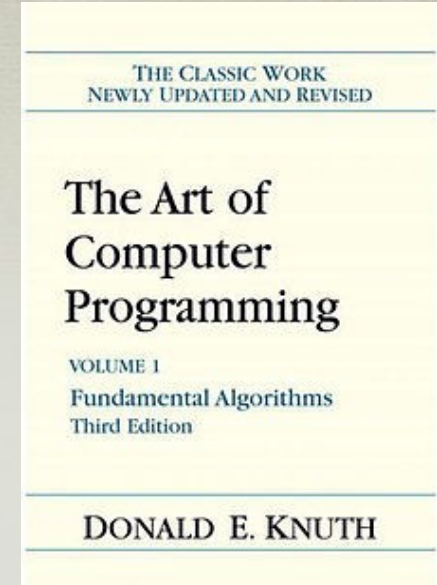
- Kako snovati algoritme?
- Predpogoj
  - dobro razumevanje problema
- Cilj
  - opis algoritma
- Kriteriji
  - pravilnost
  - učinkovitost
  - preprostost
  - implementabilnost



*Zapisal sem  
Evklidov algoritem  
za največji skupni  
delitelj dveh števil.*

# Snovanje algoritmov

- Kako snovati algoritme?
  - snovanje algoritmov je umetnost
    - TAOCP, Donald Knuth
  - Kdor hoče dobro pisati, mora veliko brati.
    - branje tujih programov
  - How to solve it?, George Polya
    1. Razumevanje problema
    2. Izdelava načrta
    3. Sledenje načrtu
    4. Pogled nazaj



# Snovanje algoritmov

- Metode snovanja algoritmov

- groba sila (*brute force*) in izčrpno preiskovanje (*exhaustive search*)
- sestopanje (*backtracking*)
- razveji in omeji (*branch & bound*)
- požrešno (*greedy*)
- deli in vladaj (*divide & conquer*)
- zmanjšaj in vladaj (*reduce & conquer*)
- pretvori in vladaj (*transform & conquer*)
- dinamično programiranje (*dynamic programming*)
- linearno programiranje (*linear programming*)
- ...



Več o metodah v nadaljevanju.

# *Snovanje algoritmov*

- Opisni jeziki
  - naravni jezik
  - diagrami poteka
  - psevdokoda
  - programski jezik
  - strojna koda
  - itd.



# Snovanje algoritmov

- Naravni jezik
  - nejasnost
  - dvoumnost
  - primeren za opis ideje

## Dvojiško iskanje elementa v urejeni tabeli

izvedemo tako, da s primerjavo iskanega elementa s sredinskim elementom tabele ugotovimo ali je iskani element v levi ali desni polovici. Nato gremo iskat element v ustrezni del tabele.


## Odločitveni ali iskalni problem

Naloga:

- urejena tabela elementov

Rešitev:

- odgovor da/ne
- indeks iskanega elementa

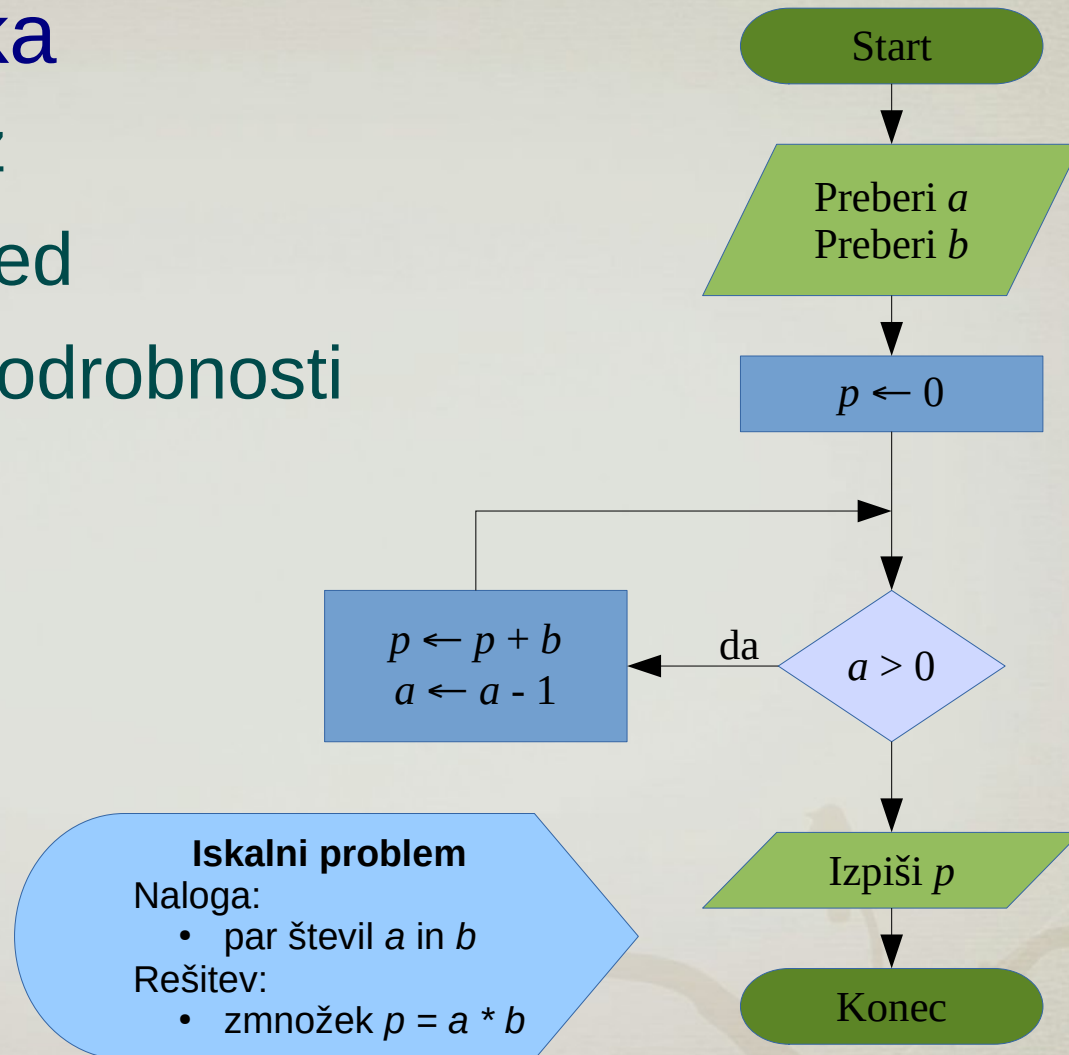


Prepisovalci bodo  
javno obešeni  
na oglasni deski.



# Snovanje algoritmov

- Diagram poteka
  - grafični prikaz
  - globalen pogled
  - okoren opis podrobnosti



# Snovanje algoritmov

- Psevdokoda

- prenosljivost?

- človek/človek
- človek/stroj, stroj/stroj

- poljubna natančnost opisa

- uporaba matematičnih formul

## Iskalni problem

Naloga:

- par števil  $a$  in  $b$

Rešitev:

- zmnožek  $p = a * b$

```
preberi a in b
```

```
 $p \leftarrow 0$ 
```

```
while a > 0 do
```

```
     $p \leftarrow p + b$ 
```

```
     $a \leftarrow a - 1$ 
```

```
endwhile
```

```
izpiši p
```

Glej tudi: [http://en.wikipedia.org/wiki/Sieve\\_of\\_Eratosthenes](http://en.wikipedia.org/wiki/Sieve_of_Eratosthenes)

Naj bo  $L$  seznam celih števil od 2 do  $N$

Ponavljaj

    Naj bo  $X$  prvi neobdelani element v seznamu  $L$

    Izpiši  $X$

    Iz seznama odstrani vse večkratnike  $X$

Dokler je  $X$  manjši od  $N$

## Iskalni problem

Naloga:

- število  $N$

Rešitev:

- seznam praštevil

# Snovanje algoritmov

- Programski jezik
  - realnost
  - ogromna izbira
  - algoritem lahko dejansko izvedemo

```
int gcd(int a, int b) {  
    if (b == 0) return a;  
    return gcd(b, a % b);  
}
```

```
gcd a 0 = a  
gcd a b = gcd b (a `rem` b)
```

```
.text  
.global pgcd  
  
pgcd:  
    push    %ebp  
    mov     %esp, %ebp  
    mov     8(%ebp), %eax  
    mov     12(%ebp), %ecx  
    push   %edx  
  
.loop:  
    cmp     $0, %ecx  
    je     .end  
    xor     %edx, %edx  
    div    %ecx  
    mov     %ecx, %eax  
    mov     %edx, %ecx  
    jmp    .loop  
  
.end:  
    pop     %edx  
    leave  
    ret
```

## Iskalni problem

Naloga:

- par števil  $a$  in  $b$

Rešitev:

- število  $g$  - gcd

```
: gcd ( a b -- c )  
    [ abs ] [ [ nip ] [ mod ] 2bi gcd ] if-zero ;
```

# Snovanje algoritmov

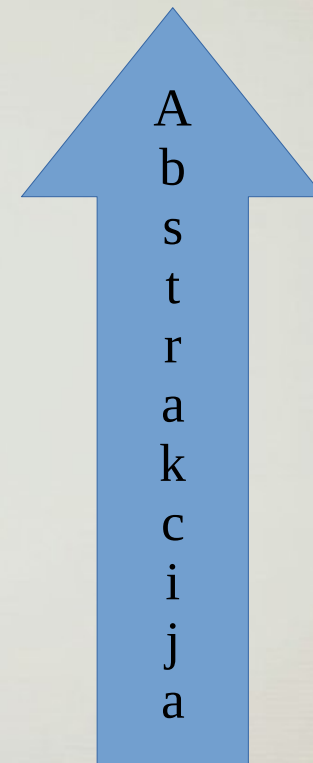
- Strojna koda
  - razumljiva le računalniku
  - hitro izvajanje

```
00000000 cf fa ed fe 07 00 00 00 01 03 00 00 80 02 00 07 00
00000010 0f 00 00 00 38 03 00 00 05 29 20 00 00 00 00 00
00000020 19 00 00 00 48 00 00 00 0f 51 30 41 47 45 a 75
00000030 52 4f 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000040 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000060 00 00 00 00 00 00 00 00 00 00 00 00 38 01 00 00
00000070 5f 5f 54 45 58 54 00 00 00 00 00 00 00 00 00 00
00000080 00 00 00 00 01 00 00 00 00 10 00 00 00 00 00 00
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000a0 07 00 00 00 05 00 00 00 03 00 00 00 00 00 00 00
000000b0 5f 5f 74 65 78 74 00 00 00 00 00 00 00 00 00 00
000000c0 5f 5f 54 45 58 54 00 00 00 00 00 00 00 00 00 00
000000d0 20 0f 00 00 01 00 00 00 00 77 00 00 00 00 00 00
000000e0 20 0f 00 00 04 00 00 00 00 00 00 00 00 00 00 00
000000f0 00 04 00 80 00 00 00 00 00 00 00 00 00 00 00 00
00001000 5f 5f 75 6e 77 69 6e 64 5f 69 6e 66 6f 00 00 00
00001100 5f 5f 54 45 58 54 00 00 00 00 00 00 00 00 00 00
00001200 98 0f 00 00 01 00 00 00 48 00 00 00 00 00 00 00
00001300 98 0f 00 00 02 00 00 00 00 00 00 00 00 00 00 00
00001400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001500 5f 5f 65 68 5f 66 72 61 6d 65 00 00 00 00 00 00
00001600 5f 5f 54 45 58 54 00 00 00 00 00 00 00 00 00 00
00001700 e0 0f 00 00 01 00 00 00 18 00 00 00 00 00 00 00
00001800 e0 0f 00 00 03 00 00 00 00 00 00 00 00 00 00 00
00001900 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00001a00 19 00 00 00 48 00 00 00 5f 5f 4c 49 4e 4b 45 44
00001b00 49 54 00 00 00 00 00 00 10 00 00 01 00 00 00 00
00001c00 00 10 00 00 00 00 00 00 10 00 00 00 00 00 00 00
00001d00 d0 00 00 00 00 00 00 07 00 00 00 01 00 00 00
00001e00 00 00 00 00 00 00 00 22 00 00 80 30 00 00 00 00
00001f00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*
00002100 00 10 00 00 38 00 00 00 02 00 00 00 18 00 00 00
00002200 58 10 00 00 04 00 00 00 98 10 00 00 38 00 00 00
00002300 0b 00 00 00 50 00 00 00 00 00 00 00 00 00 00 00
00002400 00 00 00 00 03 00 00 00 03 00 00 00 01 00 00 00
00002500 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*
```

```
0000280 0e 00 00 00 20 00 00 00 0c 00 00 00 2f 75 73 72
0000290 2f 6c 69 62 2f 64 79 6c 64 00 00 00 00 00 00 00
00002a0 1b 00 00 00 18 00 00 00 56 2d 90 64 27 5b 38 9c
00002b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00002c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00002d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00002e0 70 0f 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00002f0 0c 00 00 00 38 00 00 00 18 00 00 00 02 00 00 00
0000300 01 01 ad 04 00 00 01 00 2f 75 73 72 2f 6c 69 62
0000310 2f 6c 69 62 53 79 73 74 65 6d 2e 42 2e 64 79 6c
0000320 69 62 00 00 00 00 00 00 26 00 00 00 10 00 00 00
0000330 38 10 00 00 08 00 00 00 29 00 00 00 10 00 00 00
0000340 40 10 00 00 00 00 00 00 2b 00 00 00 10 00 00 00
0000350 40 10 00 00 18 00 00 00 00 00 00 00 00 00 00 00
0000360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*
0000f20 55 48 89 e5 48 83 ec 10 89 7d f8 89 75 f4 81 7d
0000f30 f4 00 00 00 00 0f 85 0b 00 00 00 8b 45 f8 89 45
0000f40 fc e9 14 00 00 00 8b 7d f4 8b 45 f8 99 f7 7d f4
0000f50 89 d6 e8 c9 ff ff ff 89 45 fc 8b 45 fc 48 83 c4
0000f60 10 5d c3 66 66 66 66 2e 0f 1f 84 00 00 00 00 00
0000f70 55 48 89 e5 48 83 ec 10 bf 2a 00 00 00 be 17 00
0000f80 00 00 e8 99 ff ff ff be 00 00 00 00 89 45 fc 89
0000f90 f0 48 83 c4 10 5d c3 90 01 00 00 00 1c 00 00 00
0000fa0 00 00 00 00 1c 00 00 00 00 00 00 00 1c 00 00 00
0000fb0 02 00 00 00 20 0f 00 00 34 00 00 00 34 00 00 00
0000fc0 98 0f 00 00 00 00 00 00 34 00 00 00 03 00 00 00
0000fd0 0c 00 01 00 10 00 01 00 00 00 00 00 00 00 00 01
0000fe0 14 00 00 00 00 00 00 00 01 7a 52 00 01 78 10 01
0000ff0 10 0c 07 08 90 01 00 00 00 00 00 00 00 00 00 00
00010000 00 01 5f 00 05 00 03 5f 6d 68 5f 65 78 65 63 75
00010100 74 65 5f 68 65 61 64 65 72 00 26 67 63 64 00 2a
00010200 6d 61 69 6e 00 2f 02 00 00 00 03 00 a0 1e 00 03
00010300 00 f0 1e 00 00 00 00 00 a0 1e 50 00 00 00 00 00
00010400 fa de 0c 05 00 00 00 14 00 00 00 01 00 00 00 00
00010500 00 00 00 00 00 00 00 00 02 00 00 00 0f 01 10 00
00010600 00 00 00 00 01 00 00 00 16 00 00 00 0f 01 00 00
00010700 20 0f 00 00 01 00 00 00 1b 00 00 00 0f 01 00 00
00010800 70 0f 00 00 01 00 00 00 21 00 00 00 01 00 00 01
00010900 00 00 00 00 00 00 00 00 20 00 5f 5f 6d 68 5f 65
00010a00 78 65 63 75 74 65 5f 68 65 61 64 65 72 00 5f 67
00010b00 63 64 00 5f 6d 61 69 6e 00 64 79 6c 64 5f 73 74
00010c00 75 62 5f 62 69 6e 64 65 72 00 00 00 00 00 00 00
```

# *Snovanje algoritmov*

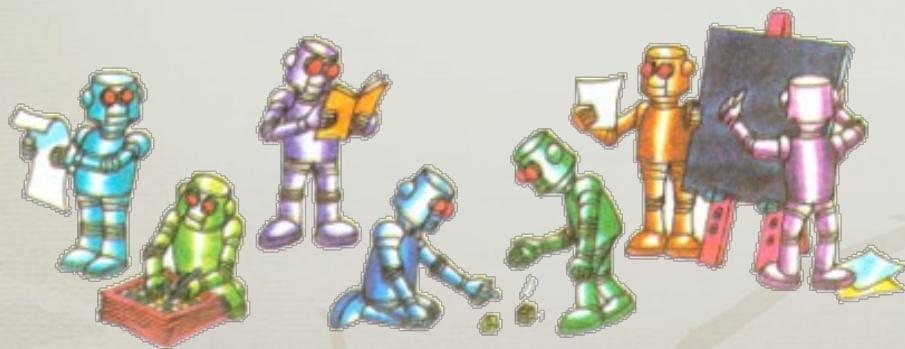
- Semantične vrzeli
  - ideja algoritma
  - ↓
  - opis algoritma (algoritem)
  - ↓
  - izvorna koda
  - ↓
  - strojna koda
  - ↓
  - proces



# Implementacija algoritmov

- Programiranje 1, 2, 3, ...
  - branje vhoda in izpis izhoda
  - aritmetične in logične operacije
  - nizi in tabele (polja, *array*)
  - odločitveni stavki
  - iteracija oz. zanke

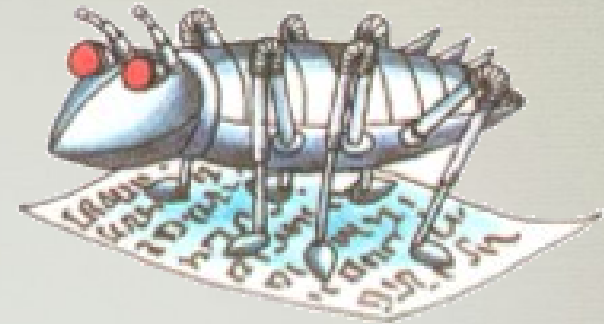
Predpostavljamo,  
da programiranje  
obvladate.



# Implementacija algoritmov

- Razhroščevanje kode

- `printf` metoda
- sledenje programu (*trace*)
- prekinitvena točka (*breakpoint*)
- opazovanje (*watch*)



- Profiliranje in instrumentacija kode

- ugotavljanje, koliko časa/pomnilnika/itd. porabijo posamezni deli programa
- programu dodamo ukaze za merjenje

# *Implementacija algoritmov*

- Izvajanje programa
  - prevajanje izvorne kode v strojno kodo
  - interpretiranje izvorne kode
- Izvedba algoritma
  - program zaženemo na ustreznem računalniku
  - izvajanje eksperimentov
  - znanstvena metoda: hipoteza
  - eksperimentalno ovrednotenje algoritma



# Sled algoritma

- Sled algoritma
  - izpis podatkov tekom izvajanja, npr.:
    - spremenljivke, podatkovne strukture
    - št. korakov, globina rekurzije, itd.
- Izvajanje
  - simuliranje na papir
  - dejansko z računalnikom



# Sled algoritma

- Evklidov algoritem

```
int gcd(int a, int b) {  
    if (b == 0) return a;  
    return gcd(b, a % b);  
}
```

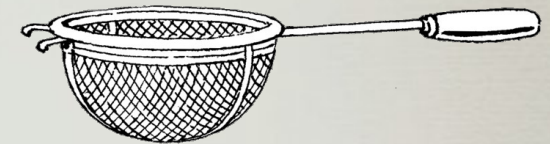
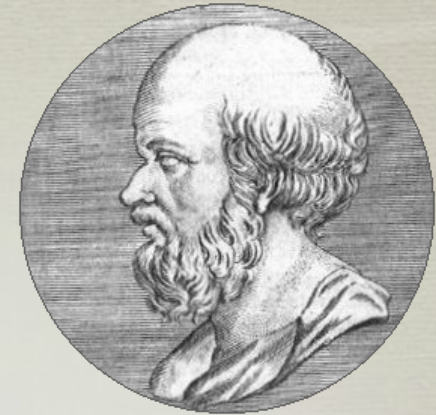
**gcd(264, 72)**

#	a	b	q	r
0	264	72	3	48
1	72	48	1	24
2	48	24	2	0
3	24	0		

**gcd(264, 72) =**  
**gcd(72, 48) =**  
**gcd(48, 24) =**  
**gcd(24, 0) = 24**

# Sled algoritma

- Eratostenovo sito



Naj bo  $L$  seznam celih števil od 2 do  $N$   
Ponavljaj  
    Naj bo  $X$  prvi neobdelani element v seznamu  $L$   
    Izpiši  $X$   
    Iz seznama odstrani vse večkratnike  $X$   
Dokler je  $X$  manjši od  $N$

Eratostenovo sito za  $n = 30$

#	izbrani $X$ in seznam $L$																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
2	2	3		5		7		9		11		13		15		17		19		21		23		25		27		29			
3	2	3		5		7				11		13				17		19				23		25					29		
4	2	3		5		7				11		13				17		19				23								29	
5	2	3		5		7				11		13				17		19				23								29	
6	2	3		5		7				11		13				17		19				23								29	
7	2	3		5		7				11		13				17		19				23								29	
8	2	3		5		7				11		13				17		19				23								29	
9	2	3		5		7				11		13				17		19				23								29	
10	2	3		5		7				11		13				17		19				23								29	
11	2	3		5		7				11		13				17		19				23								29	

# Kaj delamo pri APS?

- Razumevanje algoritma
  - psevdokoda, sled, ...
- Pravilnost algoritma
  - Ali algoritem res računa tisto, kar mislimo, da računa?
- Zahtevnost algoritma
  - Katere in koliko virov (čas, pomnilnik) potrebuje algoritem?



*Se da bolje?*

# Povzetek

- Algoritem, računski problem
  - naloga, rešitev, vrste problemov
- Snovanje algoritmov
  - opis algoritma, metode snovanja
- Implementacija algoritmov
  - eksperimenti
- Sled algoritma