



*Université de La Rochelle*



*Universiteti i Prishtinës*

# ***Algoritmat dhe kompleksiteti***

*K. Bertet, F. Berisha*

# Përkufizimi i algoritmit

- ✓ *Algoritëm*: “procedurë sistematike që prodhon – në numër të fundëm hapash – përgjegjen e një pyetjeje ose zgjidhjen e një problemi” (*Enciklopedia Britanica*)
- ✓ *Algoritëm në shkencën kompjuterike*: varg i hapave të njëkuptimtë i cili prodhon një zgjidhje të një problemi të dhënë për një kohë të fundme.
- ✓ *Program*: përshkrim kompatibil me makinën i një algoritmi të dhënë në një gjuhë programuese.

*Algoritëm*

*Reçetë kuzhine*

*Program*

*Racion*

*Programues*

*Kuzhinier*

*Makinë*

*Kuzhinë*

*Gjuhë/Sistem*

*Fultere / Pllakat për zierje*

Metafora e një  
reçete kuzhine

# Algoritmi i Euklidit

## •Teorema e Euklid-it:

$\gcd(A, B) = \gcd(B, R)$ , ku  $A > B$ ,  $B \neq 0$   
dhe  $R$  mbetja e  $A / B$ ;

$$\gcd(A, 0) = A.$$

•Algoritmi i Euklidit: skema e llogaritjes së gcd mbështetur në toeremën e Euklidit.

Algoritëm: gcd

Input: dy numra natyrorë  $A$  e  $B$ , ku  $A > B$

Output:  $\gcd(A, B)$

Përderisa  $B \neq 0$  kryej

	$R \leftarrow A \bmod B$
	$A \leftarrow B$
	$B \leftarrow R$

Kthe  $A$

# Algoritmika



- 1800 PK.: *nocioni i algoritmit*: formulim rregullash për operacione të caktuara. P.sh.: gcd.
- Shek. 9-të: termi *algoritëm* sipas një matematikani persian (al-Khowa-rismi) i cili bëri një përgjithësim përdorimi të rregullave aritmetike.
- 1930: *koncepti i algoritmit* për dallimin ndërmjet problemeve të kalkulueshme dhe problemeve të pakalkulueshme.
- 1945: gjenerata e parë e kompjuterëve elektronikë.

# Kalkulueshmëria

*Problem i kalkulueshëm:* ekziston algoritëm për zgjidhjen e tij

⇒ mund të programohet në çfarëdo makine,  
dhe në çfarëdo gjuhe programuese.

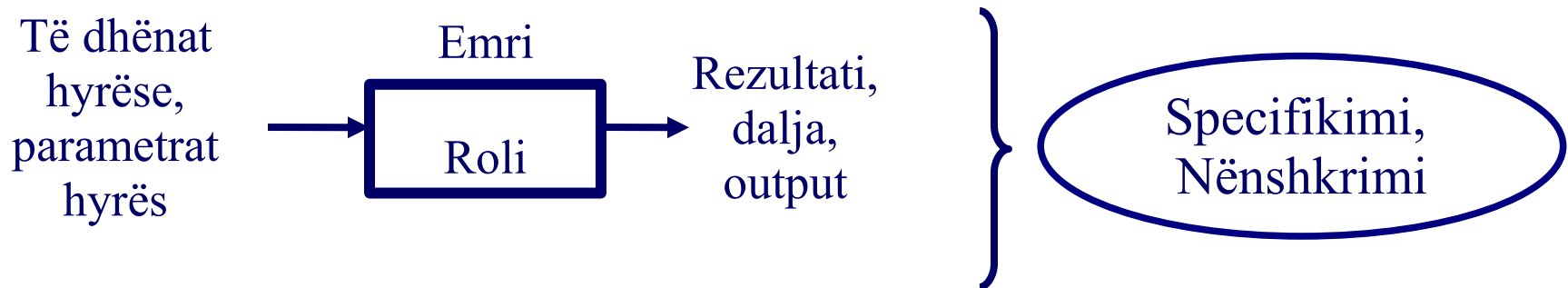
Por: një problem i kalkulueshëm,  
jo gjithmonë mund të kalkulohet në mënyrë të arsyeshme.

Shembulli i  
lojës së shahut:

- ✓ Numër i fundëm dhe i kalkulueshëm  
lëvizjesh të mundshme pas çdo dështimi
- ✓ Numër i fundëm dështimesh në një parti

⇒ Ekziston algoritëm për kalkulimin e të gjitha dështimeve të mundshme, e prandaj për të fituar (rreth  $10^{19}$  dështime të mundshme).

# Shkruarja e një algoritmi



## Përmbajtja:

- a- Instruksione ndarje vlere: variabla dhe operatorë
- b- Instruksione kushti dhe iteracioni: Në qoftë se, Përderisa, Për
- c- Funksione (metoda)

# Shkruarja e një algoritmi

- *Variablat*: për ruajtje vlerash
- *Operatorët*: mundësojnë realizimin e operacioneve elementare mbi variablat.  
mbledhje (+),                      shumëzimi (\*),                      moduo (mod),  
ndarje vlere ( $\leftarrow$  ose =),                      krahasimi (=, <, >, ...),
- *Instrukcion i thjeshtë (shprehje)*: kombinim variablash dhe operatorësh.

⇒ Çdo algoritëm mund të shkruhet me anë të instruksioneve të thjeshta dhe me anë të instruksioneve të kontrollit *Në qoftë se* dhe *Përderisa* (makina e Turing-ut).

Instrukcionet e kontrollit *Për*: avansojnë lexueshmërinë  
Funksionet (metodat): avansojnë modularitetin

# InstrukSIONET E kushtit

Algoritëm: maximum1

Input:  $A, B$

Output:  $\max(A, B)$

Në qoftë se  $A > B$  atëherë

$\max = A$

përndryshe

$\max = B$

Kthe  $\max$

Në qoftë se  $\langle kusht \rangle$   
atëherë  $\langle instruksione \rangle$   
përndryshe  $\langle instruksione \rangle$

# Instruktionet e rrjedhës së kontrollit

Algoritëm: multiplication

Input:  $A, B$

Output: prodhimi  $A*B$

Roli: llogarit  $A*B$

me anë të mbledhjeve suksesive

$mult = 0$

Përderisa  $A \geq 1$  kryej

$mult = mult + B$

$A = A - 1$

Kthe  $mult$

Përderisa  $\langle kusht \rangle$   
kryej  $\langle instruksione \rangle$

# Instruktionet e rrjedhës së kontrollit

Algoritëm: **maximum2**

Input: një koleksion  $S$  vlerash hyrëse

Output:  $\max S$

$max$  = elementi i parë i  $S$

Për  $x \in S$  kryej

    Në qoftë se  $x > max$  atëherë

$max = x$

Kthe  $max$

Për  $\langle var \rangle \in \langle koleksion \rangle$   
kryej  $\langle instruksione \rangle$

# Instruktionet e rrjedhës së kontrollit

Algoritëm: **maximum3**

Input: një varg  $T$  vlerash hyrëse

Output:  $\max_{i=1, \dots, n} T_i$

$max = T[1]$

Për  $i = 2$  deri  $n$  kryej

    Në qoftë se  $T[i] > max$  atëherë

$max = T[i]$

Kthe  $max$

Për  $\langle var \rangle$

$\langle n \rangle$  deri  $\langle m \rangle$

[me hap  $\langle numër\ i\ plotë \rangle$ ]

kryej  $\langle instruksione \rangle$

# Invokimi i një funksioni

Algoritëm: **maximum4**

Input: një varg  $T$  vlerash hyrëse

Output:  $\max_{i=1, \dots, n} T_i$

$max = T[1]$

Për  $i = 2$  deri  $n$  kryej

$max = \mathbf{maximum1}(T[i], max)$

Kthe  $max$

Invokim funksioni  
(konsistent me  
nënshkrimin e tij)

# Shkruarja e një algoritmi



- Një algoritëm i dedikohet një programuesi, e jo një makine:
  - ⇒ Jo rregulla sintaksore, por rregulla të të kuptuarit (semantike)
  - ⇒ Shkalla e përshkrimit të algoritmit varet nga programuesi (fillestar ose ekspert).
- Efikasiteti i një algoritmi mund “të matet” duke llogaritur kompleksitetin e tij.

# Kompleksiteti



- *Kompleksitet*: funksion  $O$  që vlerëson kohën e kalkulimit:

- funksion i  $n$ , vlerës së madhësisë së të dhënave hyrëse

- në *rastin më të keq*

- pavarësisht nga makina dhe gjuha programuese e implementuar

Lejon të krahasohen algoritmat, ose të vërtetohet se a është i arsyeshëm një algoritëm.

# Kompleksiteti



- *Kompleksiteti i një algoritmi*: funksion për vlerësimin e kohës së kalkulimit pavarësisht nga implementimi i tij (makinë + gjuhë).
  - ⇒ vegël për të vërtetuar se a është një algoritëm “i arsyeshëm”
  - ⇒ vegël për të krahasuar se a janë ekuivalent dy algoritma që zgjidhin të njëjtin problem (algoritma ekuivalentë)
- Kompleksiteti fitohet duke shqyrtuar:
  - një ose më tepër të dhëna hyrëse (madhësinë)
  - ciklet e futura përbrenda njëra tjetrës
  - kompleksitetin e funksioneve të përdorura

.

# Notacioni O-e-madhe



•*Përkufizim.* Për një funksion  $f(n)$  themi se është  $O(g(n))$  në qoftë se ekzistojnë konstantat  $C$  dhe  $n_0$  të tilla që për çdo  $n > n_0$

$$f(n) \leq C g(n)$$

# Kompleksiteti: shembull

Algoritëm: **maximum**

Input: një varg  $T$  prej  $n$  vlerash hyrëse

Output:  $\max_{i=1, \dots, n} T_i$

$max = T[1]$

Për  $i = 2$  deri  $n$  kryej

    Në qoftë se  $T[i] > max$  atëherë  
         $max = T[i]$

Kthe  $max$

Të dhënat hyrëse  
të madhësisë  $n$

$n$  ekzekutime të ciklit  
(iteracione)

**$O(n)$**

# Kompleksiteti

$O(1)$	Konstantë
$O(\log n)$	Logaritmik
$O(n)$	Linear
$O(n^2)$	Kuadratik
$O(n^k)$	Polinomial
$O(2^n)$	Eksponencial

I arsyeshëm

Jo i arsyeshëm  
për  $k > 4$

Jo i arsyeshëm