

Κλάσεις Πολυπλοκότητας

$f(n)=a$ (a σταθερά) πολυπλοκότητα σταθερή

$f(n)=\log(n)$ πολυπλοκότητα λογαριθμική

.....

$f(n)=n$ πολυπλοκότητα γραμμική ($an+b$)

$f(n)=n\log(n)$

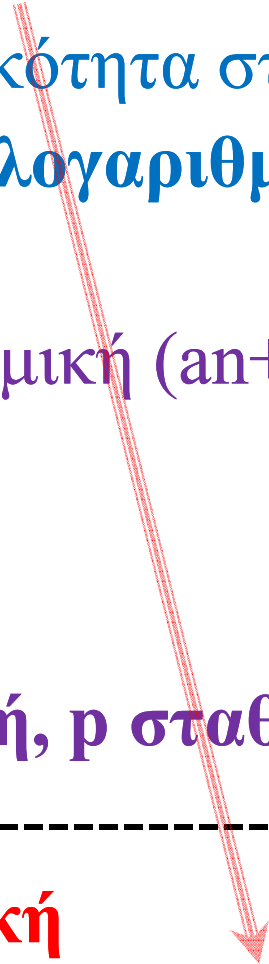
$f(n)=n^2$

.....

$f(n)=n^p$ πολ/τα πολυωνυμική, p σταθερά

$f(n)=a^n$, $1 < a$, πολ/τα εκθετική

.....



Ρυθμός αύξησης

Κάθε εκθετική συνάρτηση με βάση μεγαλύτερη από 1 αυξάνεται γρηγορότερα από κάθε πολυωνυμική συνάρτηση

Για κάθε a και b σταθερές ($a > 1$)

$$\lim_{n \rightarrow \infty} \frac{n^b}{a^n} = 0 \quad n^b = o(a^n)$$

Ρυθμός αύξησης

- $f(n)$ πολυλογαριθμικά φραγμένη αν για κάποια σταθερά k $f(n) = \log^k n$
- Κάθε θετική πολυωνυμική συνάρτηση αυξάνεται γρηγορότερα απο κάθε πολυλογαριθμική συνάρτηση.
- Για σταθερές b και a ($a > 0$)

$$\lim_{n \rightarrow \infty} \frac{\log^b n}{n^a} = 0 \quad \log^b n = o(n^a)$$

ο , ω - Συμβολισμοί

f και g θετικές συναρτήσεις

$$\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = 0 \Rightarrow g(n) = o(f(n))$$

$$o(f(n)) = \{g(n) \mid \forall c > 0, \exists n_0: \forall n > n_0, g(n) < cf(n)\}$$

$$\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = \infty \Rightarrow g(n) = \omega(f(n))$$

$$\omega(f(n)) = \{g(n) \mid \forall c > 0, \exists n_0: \forall n > n_0, g(n) > cf(n)\}$$

Ασκήσεις

✱ Ποια είναι η πολυπλοκότητα του παρακάτω προγράμματος;

```
for i:=1 to n do
  for j:=1 to  $\frac{i+1}{2}$  do
    x:=x+1;
```

✱ Να δειχτεί ότι 3^n δεν είναι $O(2^n)$

Γινόμενο Τετραγωνικών Πινάκων

Δεδομένα : $A=(a_{ij})$ και $B=(b_{ij})$ $n \times n$ πίνακες

Πρόβλημα : Να βρεθεί το γινόμενο

$$C=(c_{ij}) = \sum_{\kappa=1}^n a_{i\kappa} \cdot b_{\kappa j}$$

Γινόμενο Πινάκων(A, B:matrix, C:matrix)

for i=1 to n do

 for j=1 to n do

 c_{ij}=0;

 for k:=1 to n do

 c_{ij}:=c_{ij}+(a_{ik}·b_{kj})

Η έννοια του βέλτιστου αλγορίθμου.

Έστω πρόβλημα P

Να βρεθεί ο **καλύτερος** αλγόριθμος που το επιλύει.

- **Αλγόριθμος με την καλύτερη πολυπλοκότητα.**

“Αν οποιοσδήποτε αλγόριθμος που επιλύει το P έχει ασυμπτωτική πολ/τα $\Omega(f(n))$ τότε ο αλγόριθμος με πολ/τα $O(f(n))$ είναι **βέλτιστος (ασυμπτωτικά)**”

Γινόμενο Τετραγωνικών Πινάκων

→ n^3 πολ/μούς

→ n^2 ελ. αρ. πολ/μών

} Δεν γνωρίζουμε αλγόριθμο με $O(n^2)$

• Strassen, '69 → $O(n^{2.81})$

• Coppersmith & Winograd, '86 → $O(n^{2.379})$

Insertion Sort (A)

for **i=1** to **n** do

value = a[i]

j = i-1

while **j ≥ 1** and value > a[j] do

a[j+1] = a[j] * ολίσθηση

j = j-1

endwhile

a[j+1] = value * εισαγωγή

endfor

Bubble Sort (A)



```
for i=n to 1 do
```

```
  for j=1 to i-1 do
```

```
    if  $a[j+1] > a[j]$  then  
      swap ( $a[j+1]$ ,  $a[j]$ )
```

```
    endif
```

```
  endfor
```

```
endfor
```

Πολυπλοκότητα

Bubble Sort (μέθοδος της φυσαλλίδας): $\Theta(n^2)$

στη χειρίστη περίπτωση (worst case)
στην κατά μέσο όρο (average case)

Insertion Sort (μέθοδος εισαγωγής) : $\Theta(n^2)$

στη χειρίστη περίπτωση (worst case)
στην κατά μέσο όρο (average case)

Ταξινόμηση n ακεραίων αριθμών
με συγκρίσεις των στοιχείων ανά 2

Το πλήθος συγκρίσεων $\geq n \log_2 n$

Αρα $\Omega(n \log_2 n)$

Quick Sort (βέλτιστος στην κατά μέσο όρο πολ/τα)

ΟΧΙ όμως και στην χειρίστη περίπτωση

Merge 2 ταξ/νες ακολουθίες

- Merge (a,b,c)

i=1;

j=1;

for k=1 to m+n do

if a[i] <= b[j]

then c[k]=a[i]; i=i+1

else c[k]=b[j]; j=j+1

Παράδειγμα: a: 2 6 7 9

b: 4 5 8

c: 2 4 5 6 7 8 9 (έξοδος)

Πομπλοκότητα: $O(m+n)$

Αλγόριθμος του Ευκλείδη

Να υπολογιστεί η ασυμπτωτική πολυπλοκότητα του ακόλουθου αλγόριθμου για την εύρεση του μέγιστου κοινού διαιρέτη δυο θετικών ακέραιων αριθμών.

$$\text{gcd}(126,48)=6,$$

$$\text{gcd}(89,55)=1, \text{gcd}(13,8)=1, \text{gcd}(12,18)=6$$

ΜΚΔ(m, n: integer): integer;

repeat

r := m mod n;

if r ≠ 0

then m := n; n := r;

endif

until r = 0;

ΜΚΔ = n

end*****

Ασκήσεις:

- 1) Για $x > -1$ έχουμε $x/(1+x) \leq \ln(1+x) \leq x$
- 2) Πολυπλοκότητα Selection-Sort(A)

Αναζήτηση ενός στοιχείου x μέσα σε ένα πίνακα S με n στοιχεία

`Search (S , x) : integer ;`

`i := 1 ;`

`an+1 := x ;`

`while ai ≠ x do i := i + 1 ;`

`Search := i ;`

`if i ≤ n then "στην i-θέση"`

`else "δεν υπάρχει"`

$C_x(n) = O(n)$ στην χειρίστη περίπτωση

Αναζήτηση ενός στοιχείου x μέσα σε ένα πίνακα S με n στοιχεία Μέση πολυπλοκότητα

Κατά μέσο όρο πολυπλοκότητα της αναζήτησης στο S :

→ Υπόθεση : $q = P[x \in S]$

$$\text{Αν } x \in S \Rightarrow P[x \text{ στην } i \text{ θέση}] = q/n$$

→ D : σύνολο δεδομένων

$$\rightarrow C_{\mu}(n) = \sum_{d \in D} P(d) \text{ cost}(d)$$

Μέση πολυπλοκότητα

Έστω $D_i = \{d \mid x \in S, i\text{-θέση}\}, D_o = \{d \mid x \notin S\}$

$$\left(\bigcup_{i=1}^n D_i\right) \cup D_o = D,$$

$D_i \cap D_j = \text{κενό για κάθε ζευγάρι } ij,$

$D_i \cap D_o = \text{κενό για κάθε } i$

$$\text{Cost}(d) = i, d \in D_i$$

$$\text{Cost}(d) = n+1, d \in D_o$$

$$C_\mu(n) = \sum_{d \in D_o} P(d)C(d) + \sum_{i=1}^n \left(\sum_{d \in D_i} P(d)C(d) \right)$$

Μέση πολυπλοκότητα

$$\begin{aligned}C_{\mu}(n) &= \sum_{d \in D_0} P(d)C(d) + \sum_{i=1}^n \left(\sum_{d \in D_i} P(d)C(d) \right) \\ &= (n+1) \sum_{d \in D_0} P(d) + \sum_{i=1}^n \left(i \sum_{d \in D_i} P(d) \right) \\ &= (n+1)(1-q) + (q/n) \sum_{i=1}^n i = (n+1)(1-q/2)\end{aligned}$$

→ $q=1$?

→ $q=1/2$?