ο , ω - Συμβολισμοί

f και g θετικές συναρτήσεις

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

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

ii. $$\lim_{n \to \infty} \frac{g(n)}{f(n)} = \infty \Rightarrow g(n) = \omega(f(n))$$

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

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

$f(n)=a^n$, $1<a$

………………………..

- **Ποια είναι η πολυπλοκότητα του παρακάτω προγράμματος;**

```
for i:=1 to n do
 for j:=1 to (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

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

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)
```

for i=1 to n do

for j=1 to n do

$c_{ij}=0;$

for k:=1 to n do

$c_{ij}:=c_{ij}+(a_{ik} \cdot b_{kj})$

Ρυθμός αύξησης

- Κάθε εκθετική συνάρτηση με βάση μεγαλύτερη απο 1 αυξάνεται γρηγορότερα απο κάθε πολυωνυμική συνάρτηση.

Για κάθε a και b σταθερές (a > 1)

$$\lim_{n \to \infty} \frac{n^b}{a^n} = 0$$

$$n^b = o(a^n)$$

Ρυθμός αύξησης

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

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

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**

$\left.\begin{array}{l} \textbf{n}^3 \text{ πολ/μούς} \\ \textbf{n}^2 \text{ ελ. αρ. πολ/μών} \end{array}\right\}$ **Δεν γνωρίζουμε αλγ. με $O(n^2$**

- Strassen, '69 $\longrightarrow O(n^{2.81})$
- Coppersmith & Winograd, '86 $\longrightarrow O(n^{2.379})$

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): Θ(n^2)**

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

- **Insertion Sort (μέθοδος εισαγωγής) : Θ(n^2)**

  στη χείριστη 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 χείριστη περίπτωση

Αλγόριθμος του Ευκλείδη

Να υπολογιστεί η ασυμπτωτική πολυπλοκότητα του ακόλουθου αλγόριθμου για την εύρεση του μέγιστου κοινού διαιρέτη δυο θετικών ακέραιων αριθμών. gcd(126,48)=6, gcd(89,55)=1, gcd (13,8)=1, 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 χ>-1 ἐχουμε $x/(1+x) \leq \ln(1+x)) \leq x$

2) Πολυπλοκότητα Selection-Sort(A)

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

```
Search(S,x):integer;

i:=1;
a_{n+1} :=x;
while a_i≠x do i:=i+1;
Search:=i;
if i≤n then "στην i-θέση"
else "δεν υπάρχει"
```

Συγκρίσεις: (n+1) + 1 ,$C_x(n) = O(n)$ στην χείριστη περίπτωση

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

Κατά μέσο όρο πολυπλοκότητα της αναζήτησης στο S:

→ Υπόθεση : Αν $x \in S \Rightarrow P[x$ στην i θέση$] = q/n$ ίδια

→ $q = P[x \in S]$

→ D: σύνολο δεδομένων

→ $C_\mu(n) = \sum_{d\ e\ D} P(d)\ \mathrm{cost}(d)$

Μέση πολυπλοκότητα

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

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

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

$D_i \cap Do$ = κενό για κάθε i

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

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

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

$$=(n+1)\sum_{d \in Do} P(d) + \sum_{i=1}^{n}\left(i \sum_{d \in Di} P(d)\right)$$

$$=(n+1)(1-q)+(q/n)\sum_{i=1}^{n} i=(n+1)(1-q/2)$$

→q=1 ….?

→q=1/2 ….?

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)

Πολυπλοκότητα: O(m+n)