



Πληροφορική

Ενότητα 7: Α. Πληροφορική: Ακολουθία Fibonacci- Αναδρομικότητα- Πολυπλοκότητα.

Β. Το πρόβλημα του περιοδεύοντος πωλητή.

Κωνσταντίνος Καρατζάς
Τμήμα Μηχανολόγων Μηχανικών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

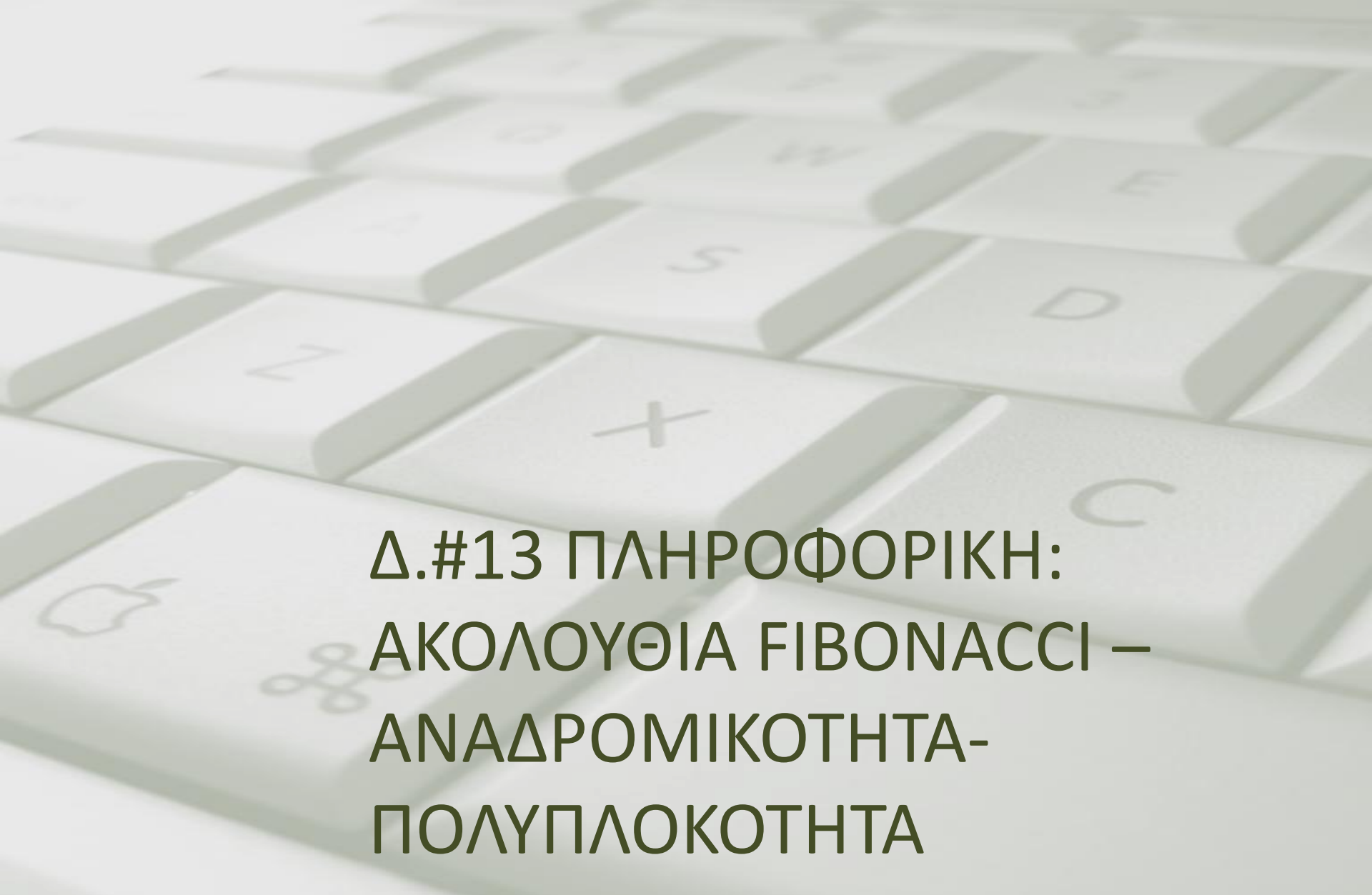
- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





Δ.#13 ΠΛΗΡΟΦΟΡΙΚΗ: ΑΚΟΛΟΥΘΙΑ FIBONACCI – ΑΝΑΔΡΟΜΙΚΟΤΗΤΑ- ΠΟΛΥΠΛΟΚΟΤΗΤΑ

21.04.2015

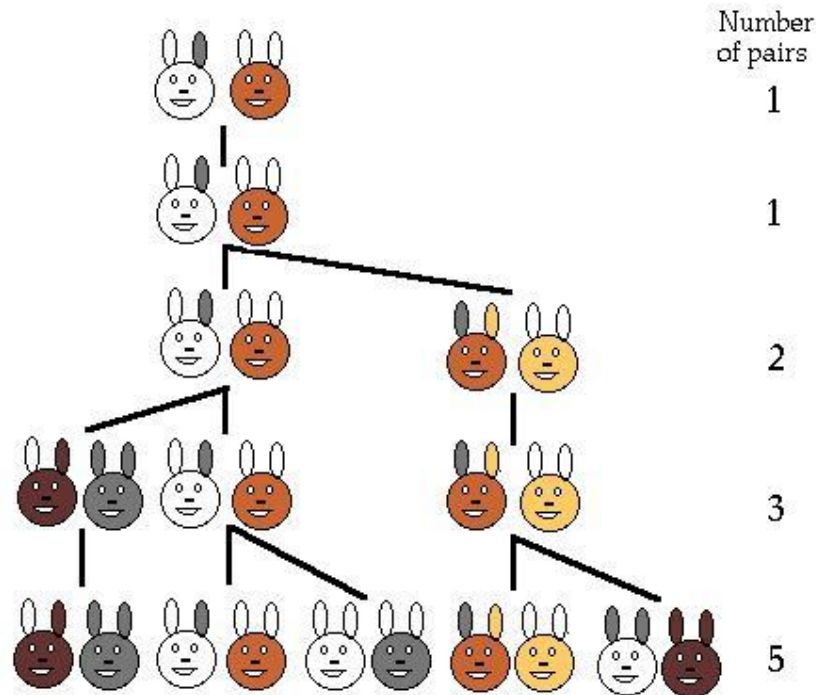
Αναπλ. Καθηγητής Κωνσταντίνος Καρατζάς
Τμήμα Μηχανολόγων Μηχανικών ΑΠΘ

Και για σήμερα...



Μία διάσημη ακολουθία με σημείο εκκίνησης την
..... κονικλοτροφία!

Αναπαράγωγή λαγών..



Ο Leonardo Pisano (Leonardo Fibonacci = Leonardo, γιος του Bonaccio, 1175-1250) πρότεινε την ομώνυμη ακολουθία το 1202 στο «*The Book of the Abacus*».

Αριθμοί Fibonacci

- Η σειρά ακεραίων 1,1,2,3,5,8,13,21,34,... που ορίζεται ως $F_{n+2} = F_n + F_{n+1}$ είναι η περίφημη **ακολουθία Fibonacci**.

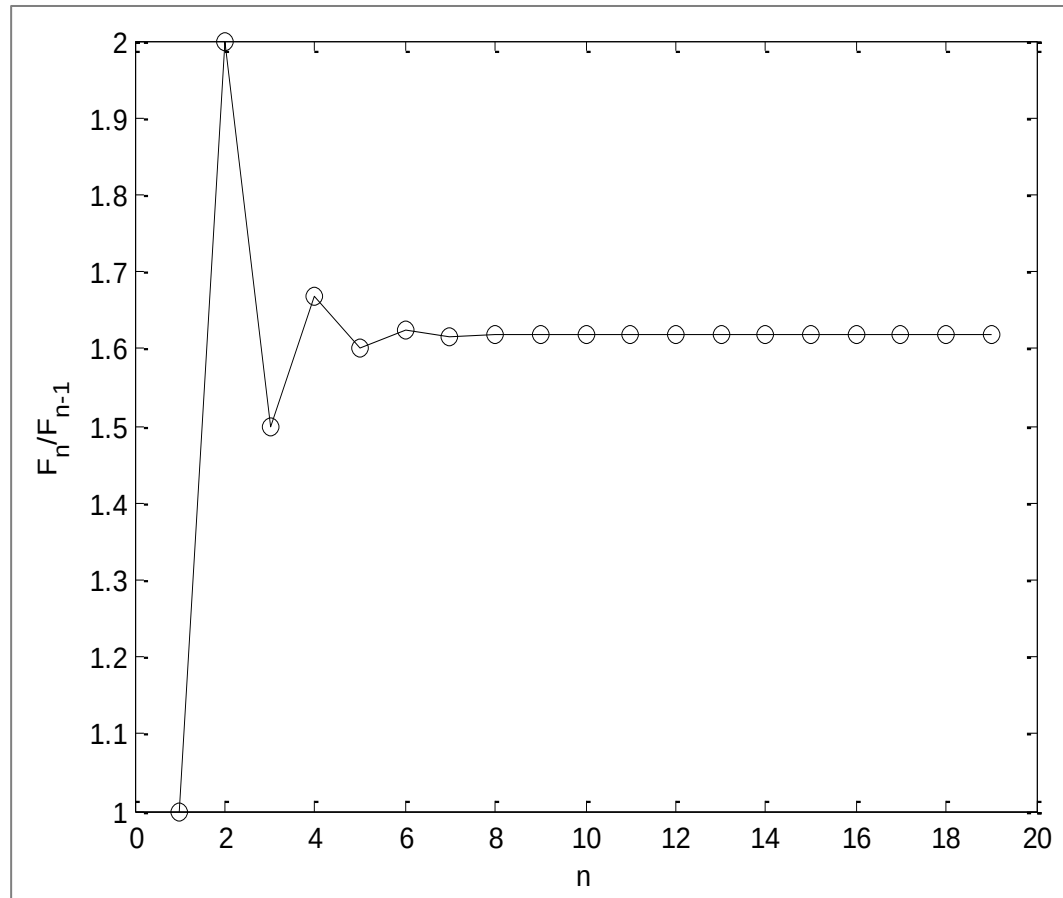
- Προσεγγιστική σχέση F_n

$$F_n = \frac{1}{\sqrt{5}} \left(\frac{1 + \sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1 - \sqrt{5}}{2} \right)^n$$



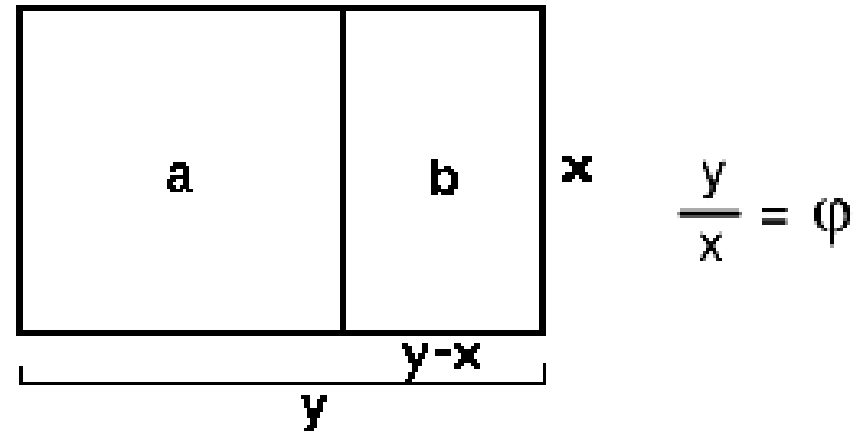
- Για μεγάλες τιμές του n , ο ρυθμός ανάπτυξης είναι $F_n \approx 0.447214 \times 1.61803^n$.
- Ο όρος $(1+\sqrt{5})/2 \approx \mathbf{1.61803}$ είναι η **χρυσή αναλογία**.

$$F_n/F_{n-1} = \phi$$



Χρυσός κανόνας

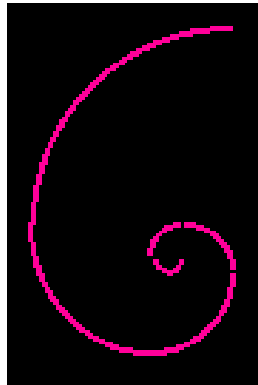
- Ένα ορθογώνιο του οποίου οι πλευρές ακολουθούν τον χρυσό κανόνα ονομάζεται χρυσό ορθογώνιο.
- Όταν ένα χρυσό ορθογώνιο “τετραγωνίζεται”, η περιοχή που “περισσεύει” σχηματίζει επίσης ένα χρυσό ορθογώνιο!



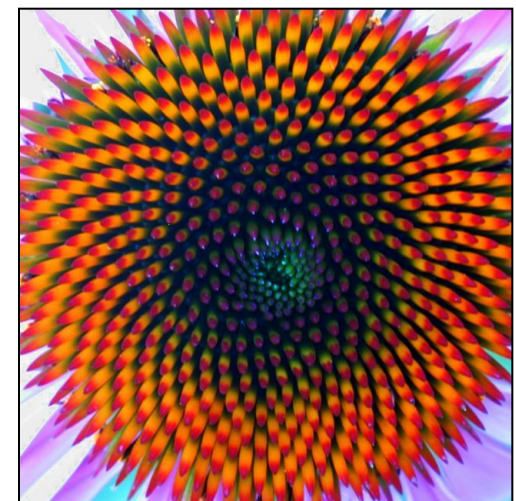
Fibonacci στη Φύση..

Οι αριθμοί Fibonacci απαντώνται συχνά στον φυσικό κόσμο.

Σχήμα κελύφων:

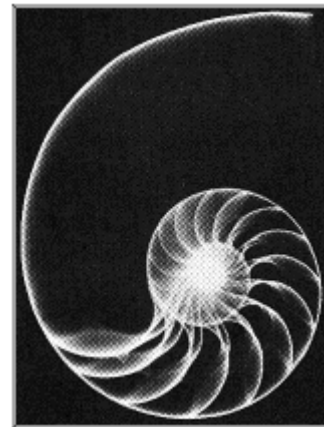


Αριθμός πετάλων σε λουλούδια



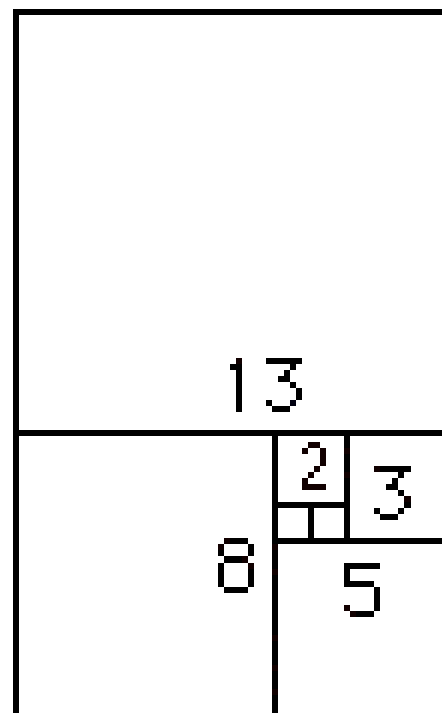
Πρόκειται για την ακολουθία υπ'αριθμ. A000045 στην On-Line Εγκυκλοπαίδεια ακεραίων ακολουθιών
<http://www.research.att.com/~njas/sequences/index.html>

Η σπείρα



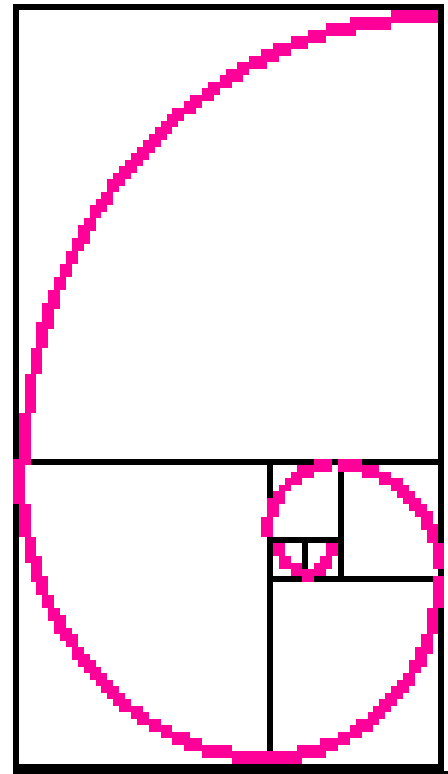
Κατασκευή...

- Τοποθετούμε **τετράγωνα** το ένα δίπλα στο άλλο με σχετικά μήκη που αντιπροσωπεύουν όρους της ακολουθίας



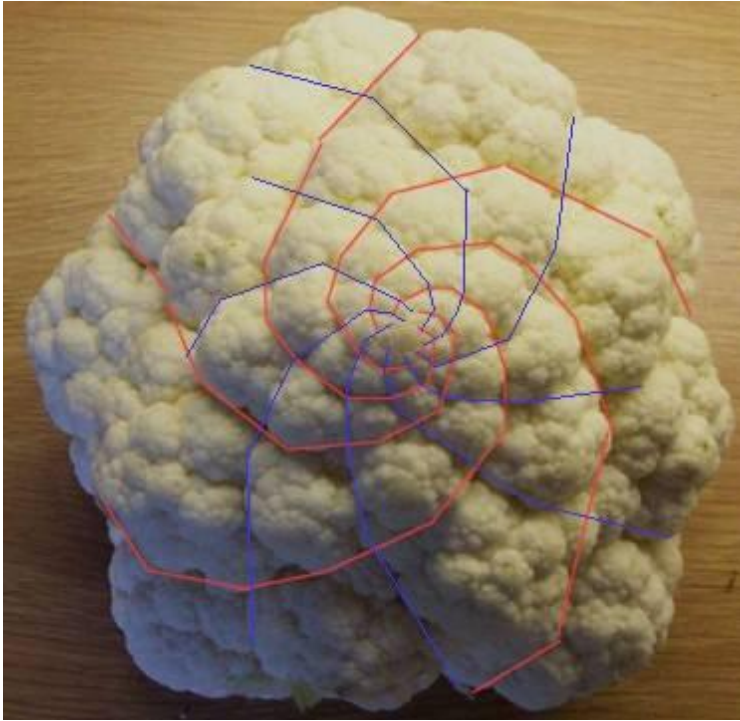
Κατασκευή...

- Ενώνουμε τις γωνίες.

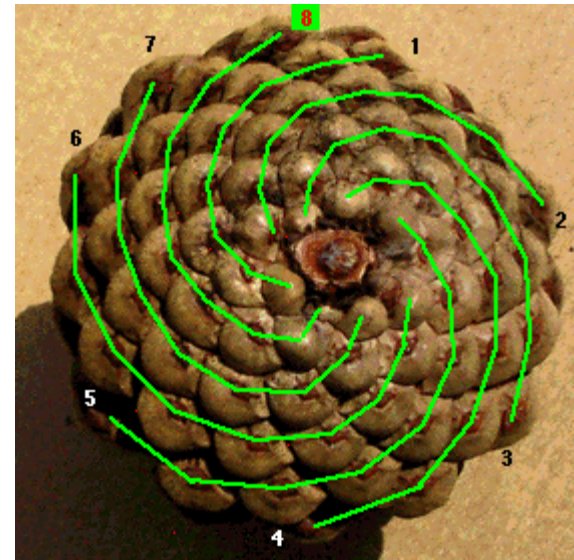


Σπείρες στη Φύση

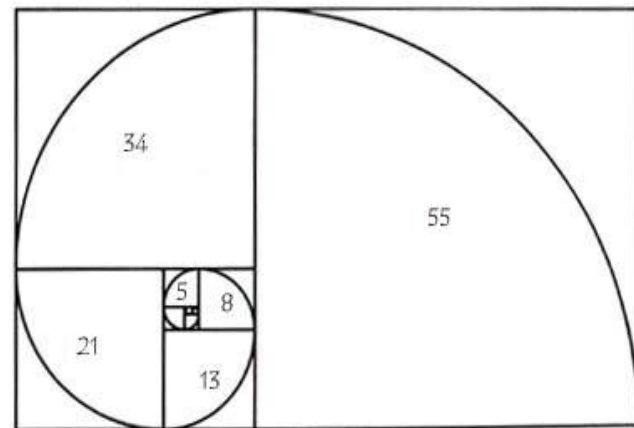
Κουνουπίδι



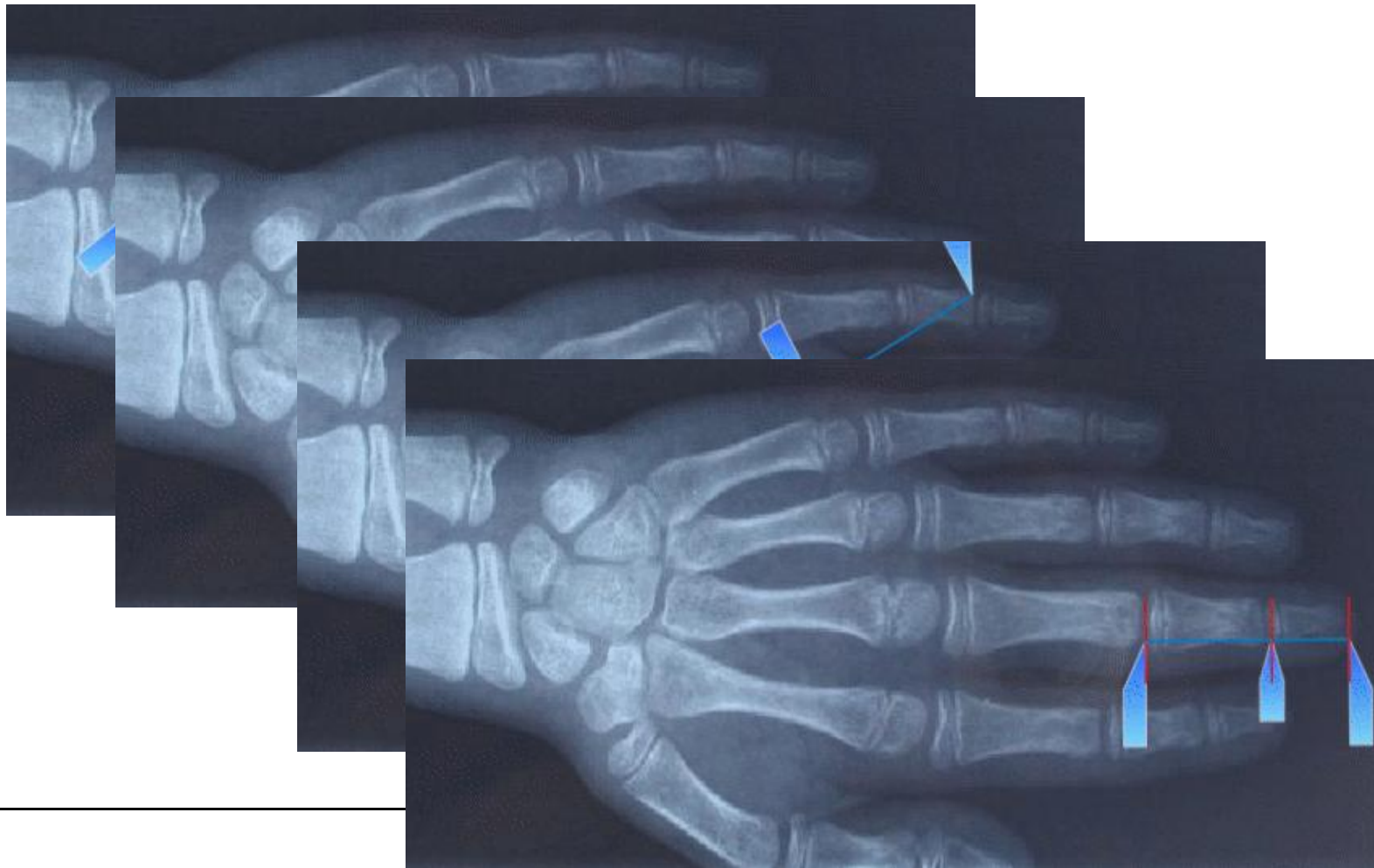
Κουκουνάρι



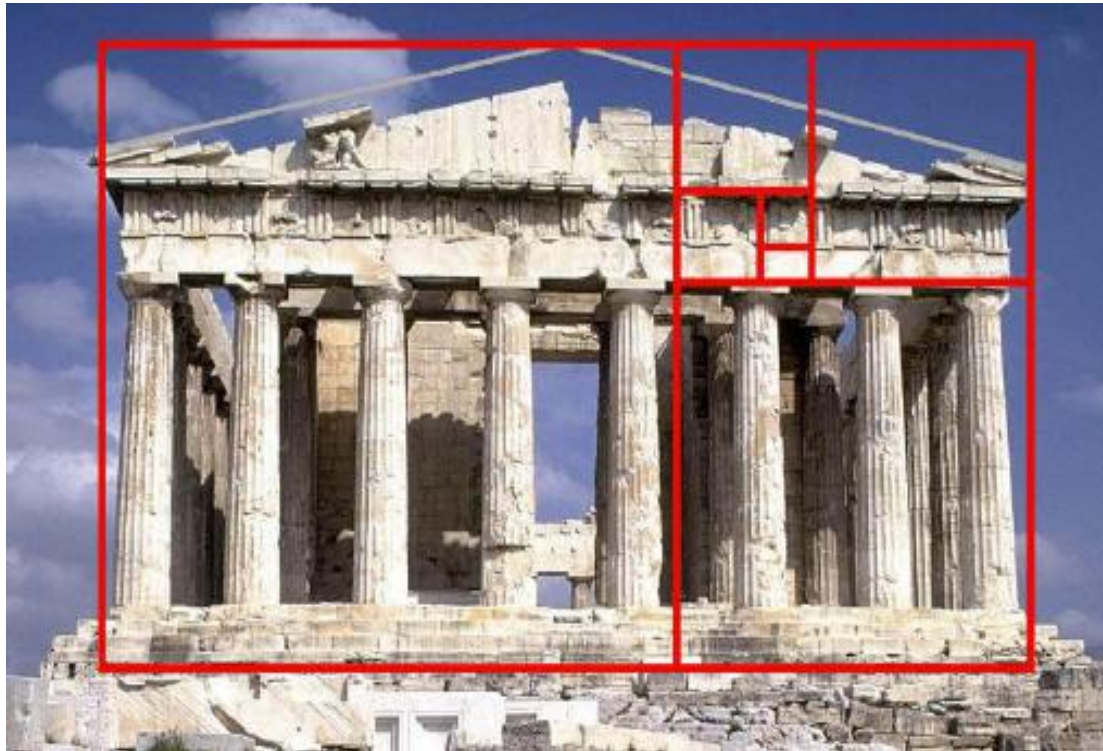
Στη φύση



Στο ανθρώπινο σώμα



Αρχιτεκτονική



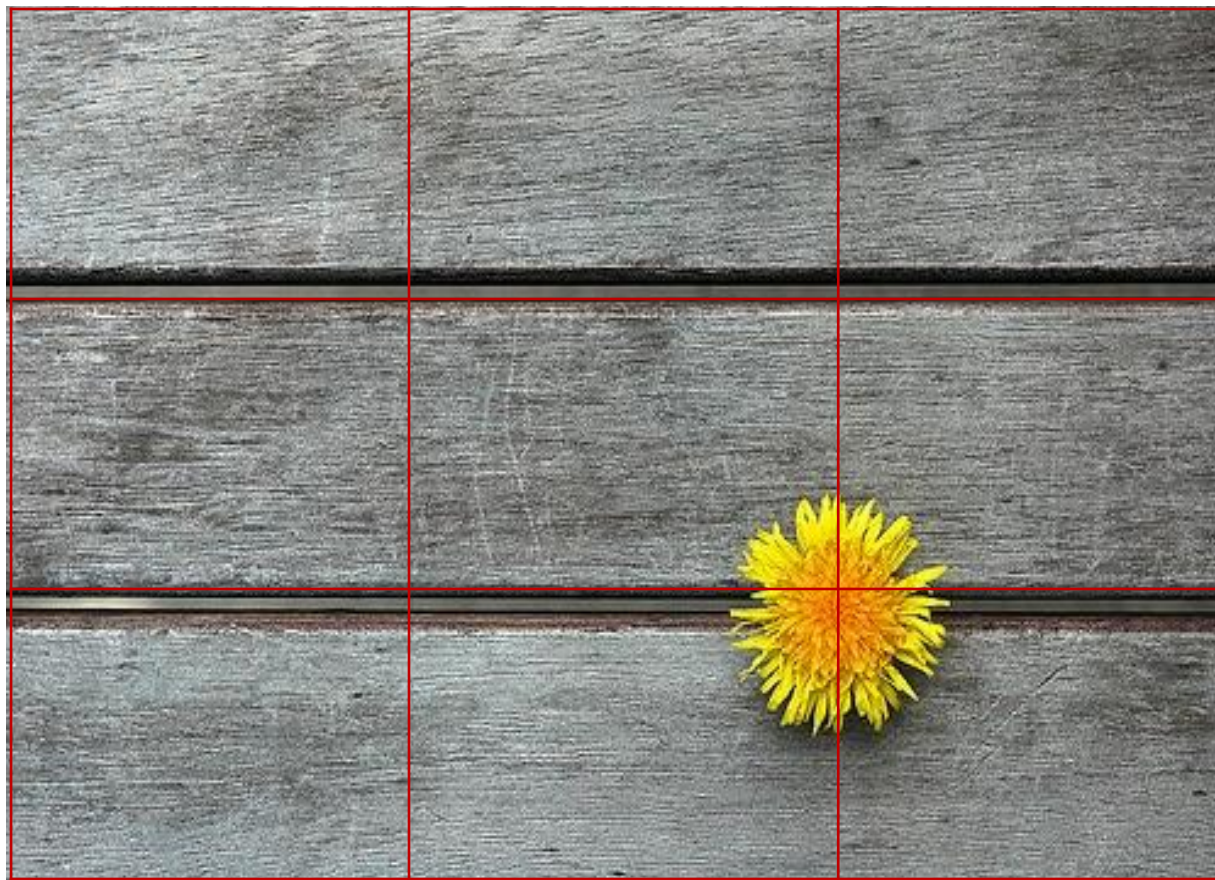
Εικόνα Α



Εικόνα Β



Κανόνας τρίτων



Κοι πως υπολογίζεται?

- Αναδρομική ακολουθία!

Αναδρομικός-ή-ό

1. που συμβαίνει, που γίνεται στο παρόν, καλύπτει όμως και μια προηγούμενη χρονική περίοδο: *Ο νόμος (δεν) έχει αναδρομική ισχύ, (δεν) ισχύει και για χρονικό διάστημα πριν από τη θέσπισή του. Θα γίνει αναδρομική αύξηση των μισθών από τον Οκτώβριο του προηγούμενου έτους. Έγινε αναδρομική έκθεση του ζωγράφου με έργα παλαιότερης δουλειάς του. || (ως ουσ.) τα αναδρομικά, χρήματα που προέρχονται από αναδρομική αύξηση: Η αξία των αναδρομικών εξανεμίζεται από την άνοδο του πληθωρισμού.*

2. (γλωσσ.) ~ σχηματισμός, για λέξη που σχηματίζεται από άλλη λέξη αναλογικά προς αντίστοιχο σχήμα, δίνει όμως την εντύπωση πως είναι η βάση του σχηματισμού, π.χ.: αλμύρα < αλμυρός, αναδομώ < αναδόμηση.
αναδρομικά ΕΠΙΡΡ: *Θα πάρουμε την αύξηση ~ από τον περασμένο Μάρτιο.*

Αναδρομικοί αλγόριθμοι

- Οι αλγόριθμοι που “χρησιμοποιούν” τον εαυτό τους για την επίλυση ενός προβλήματος (άρα “επαναλαμβάνονται”)

- Βασική περίπτωση

- Επαγωγική περίπτωση

- Έστω ότι δεν γνωρίζουμε την ύψωση σε δύναμη αλλά μόνο τον πολλαπλασιασμό

$$b^n = \begin{cases} 1, & n=0 \\ b \cdot b^{n-1}, & n>0 \end{cases}$$

- Για $n=3$

$$b^3 = b^{3-1}b = b^2b = b^{2-1}bb = bbb$$

Υλοποίηση

```
function timi = dinami(number,power)

if (power==1)
    timi=number;
else
    timi=number*dinami (number, (power-1) ) ;
end
```

Υλοποίηση



```
function timi = dinami2(number,power)
```

```
timi=number^power
```

- 
- Και πόσο χρόνο χρειάζεται ο υπολογισμός?
-

Παράδειγμα1: Υλοποίηση

```
function timi = dinam1(number,power)
tic
if (power==1)
    timi=number;
else
    timi=number*dinam1(number, (power-1)) ;
end
toc
```

```
dinam1(2,100)
```

```
Elapsed time is 0.001169 seconds.
```

```
Elapsed time is 0.001180 seconds.
```

```
Elapsed time is 0.001191 seconds.
```

```
ans = 1.2677e+030
```

Υλοποίηση

```
function timi = dinami2(number,power)
```

```
tic
```

```
Timi=number^power
```

```
toc
```

```
Elapsed time is 0.000080 seconds.
```

```
Timi =
```

```
1.2677e+030
```

Fibonacci

$$f(n) = \begin{cases} 0, & n=0 \\ 1, & n=1 \\ f(n-1) + f(n-2), & n>1 \end{cases}$$



□ Πώς θα την υλοποιούσατε?

Fibonacci

```
function fn = fibonacci1(n)

if (n==0)
    fn = 0;
elseif (n==1)
    fn = 1;
else
    fn = fibonacci1(n-2)+fibonacci1(n-1);
end
```

Fibonacci

```
function fn = fibonacci2(n)
```

```
fn(1) = 1;
```

```
fn(2) = 1;
```

```
for i=3:n
```

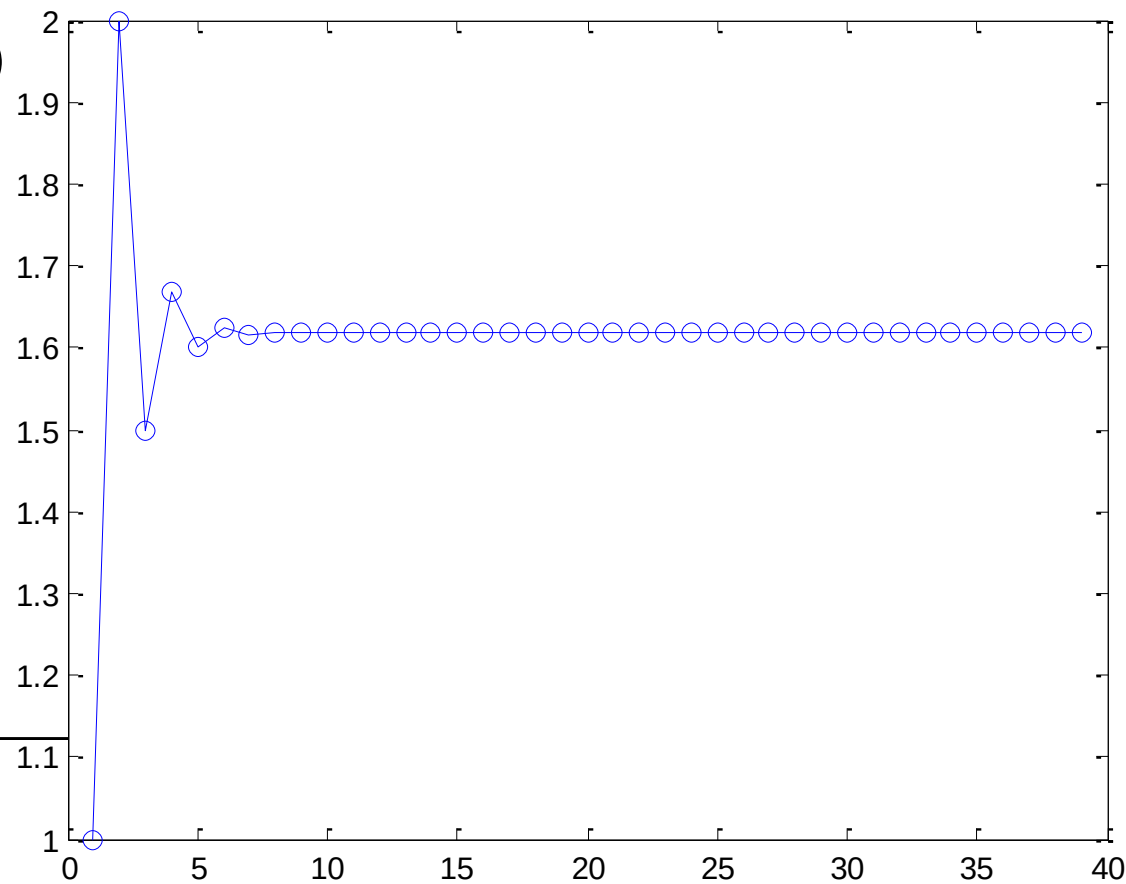
```
    fn(i) = fn(i-1) + fn(i-2);
```

```
end
```

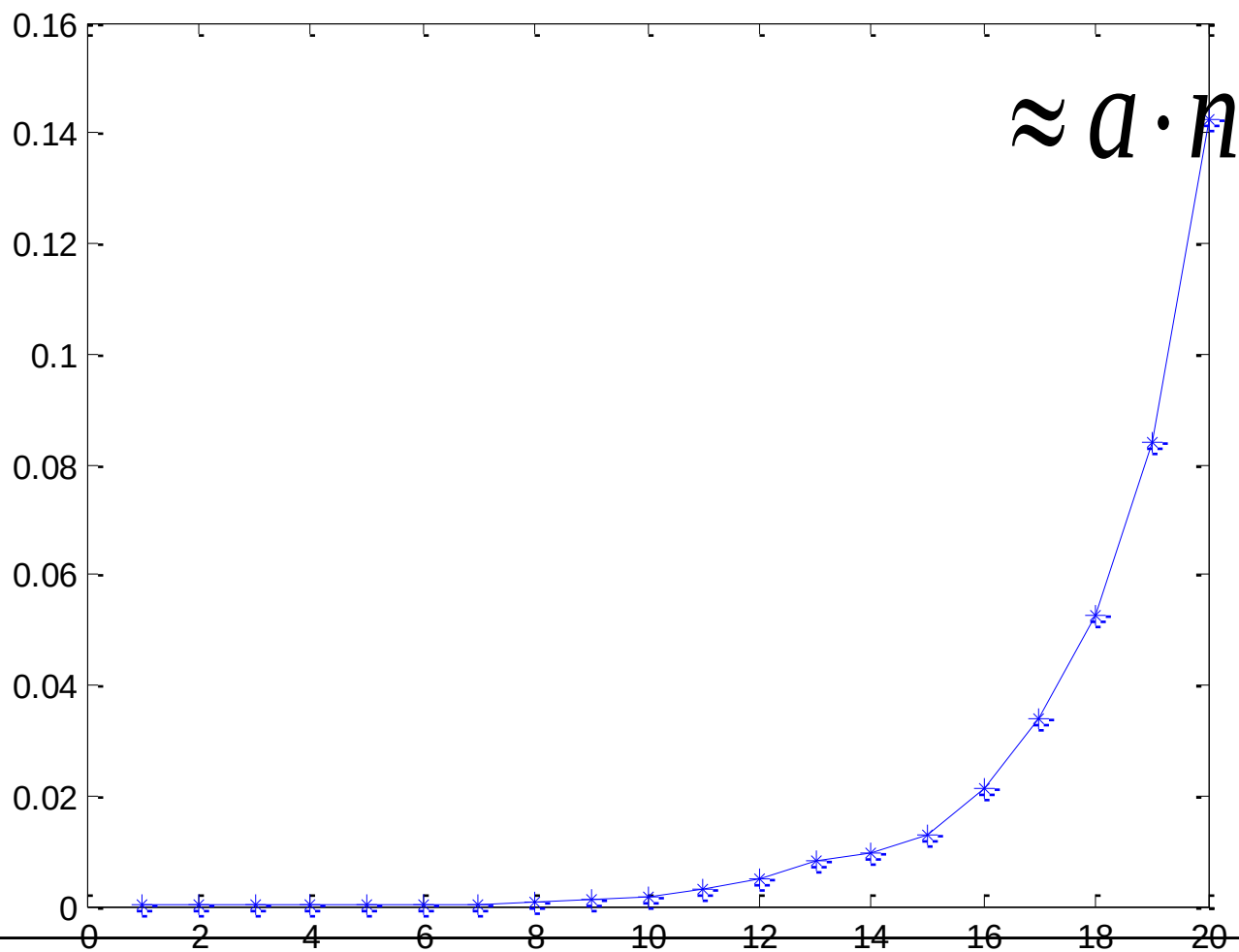
Χρυσή αναλογία

$$\lim_{n \rightarrow \infty} \frac{F(n)}{F(n-1)} = \phi \cong 1.618$$

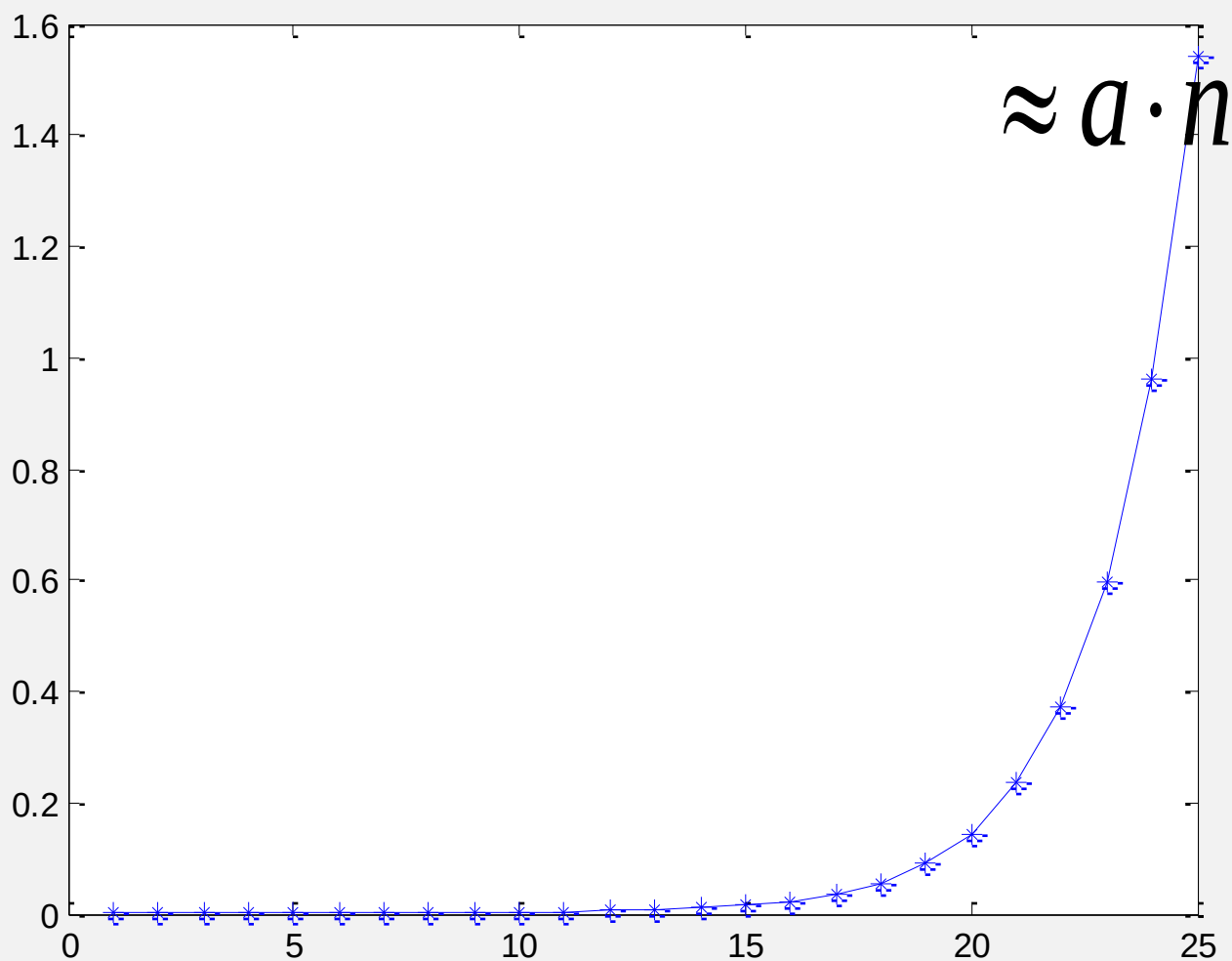
- `f=fibo(40)`
- `R=f(2:40)./f(1:39)`
- `plot(R,'b-o')`



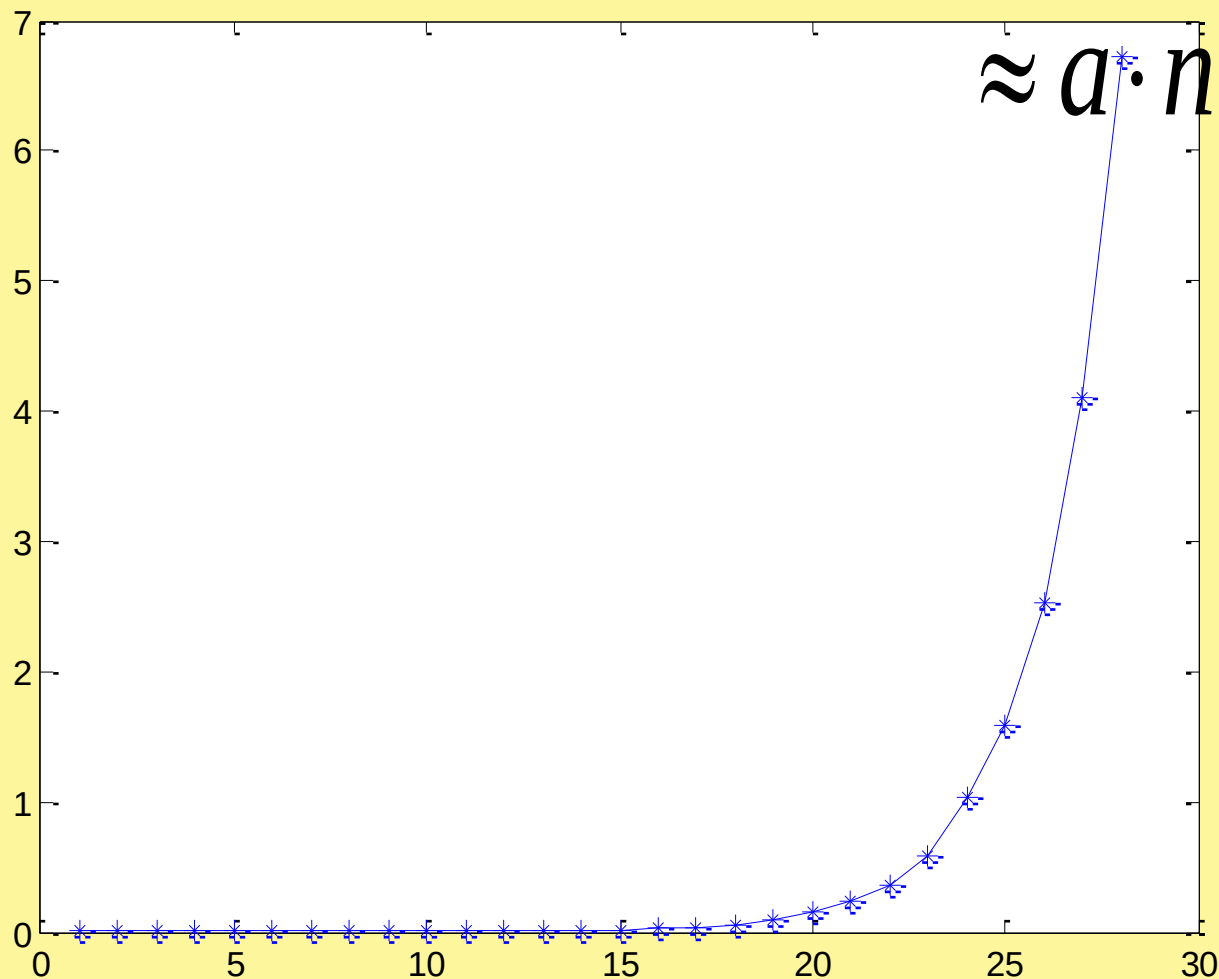
Και πόσος χρόνος απαιτείται;



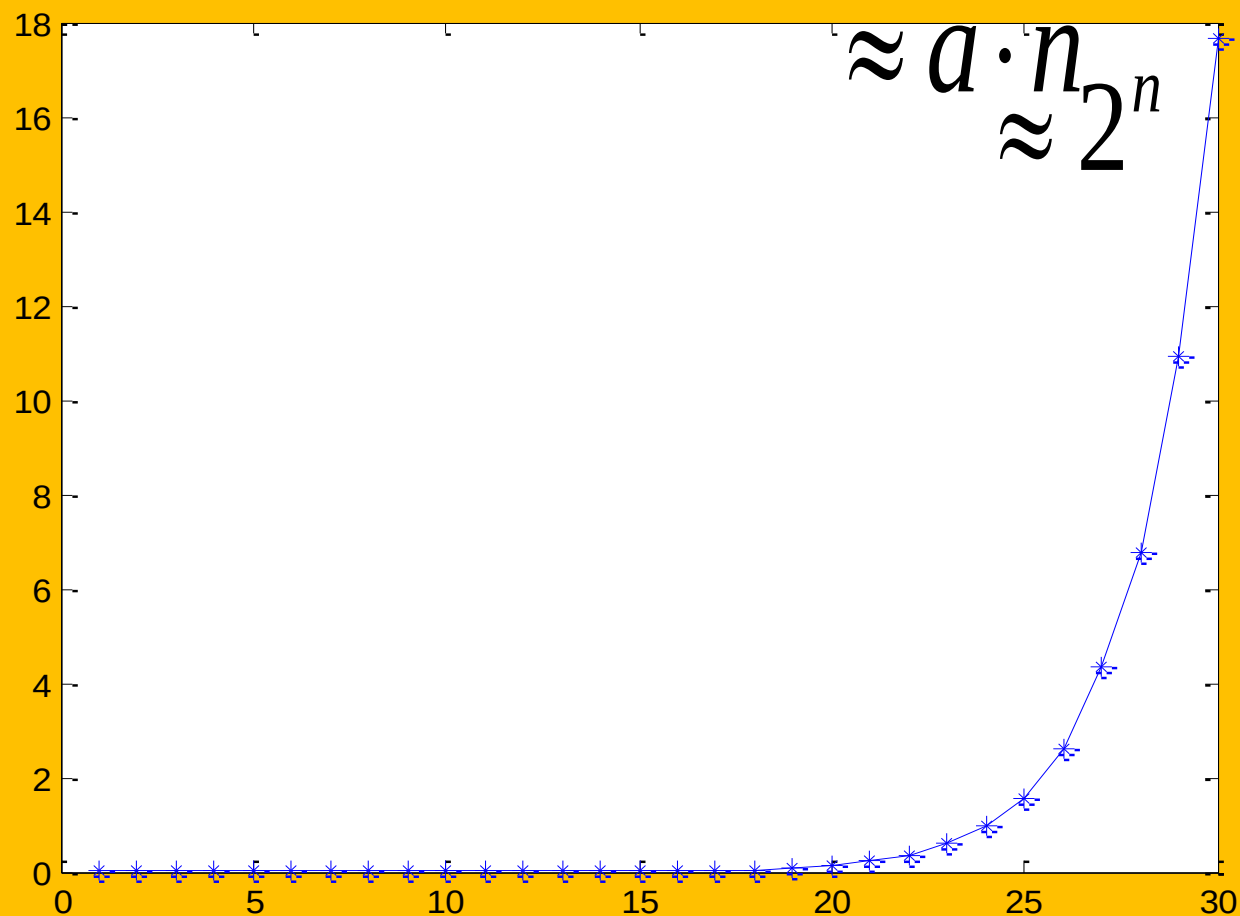
Και πόσος χρόνος απαιτείται;



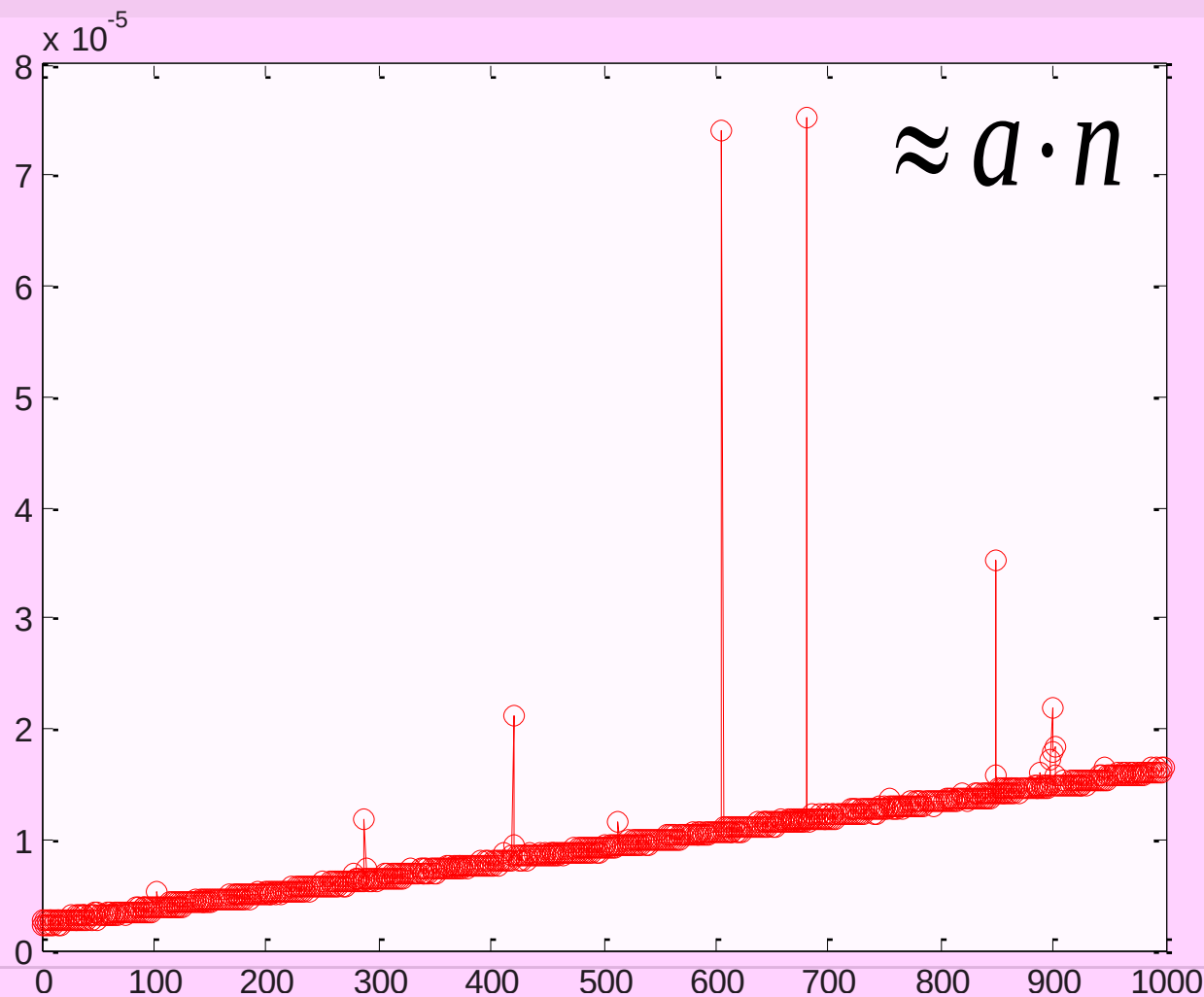
Και πόσος χρόνος απαιτείται;



Και πόσος χρόνος απαιτείται;



Και πόσος χρόνος απαιτείται;



Πώς προέκυψε αυτό?

```
function fn = fibonacci2(n)
```

```
    if (n==0)
```

```
        fn = 0;
```

```
    elseif (n==1)
```

```
        fn = 1;
```

```
    else
```

```
        f0 = 0;
```

```
        f1 = 1;
```

```
        for i=2:n
```

```
            fn = f1 + f0;
```

```
            f0 = f1;
```

```
            f1 = fn;
```

```
        end
```

```
    end
```

Πως προέκυψε αυτό?

```
function fn = fibonacci2(n)
```

```
if (n==0)
```

```
    fn = 0;
```

```
elseif (n==1)
```

```
    fn = 1;
```

```
else
```

```
    f0 = 0;
```

```
    f1 = 1;
```

```
    for i=2:n
```

```
        fn = f1 + f0;
```

```
        f0 = f1;
```

```
        f1 = fn;
```

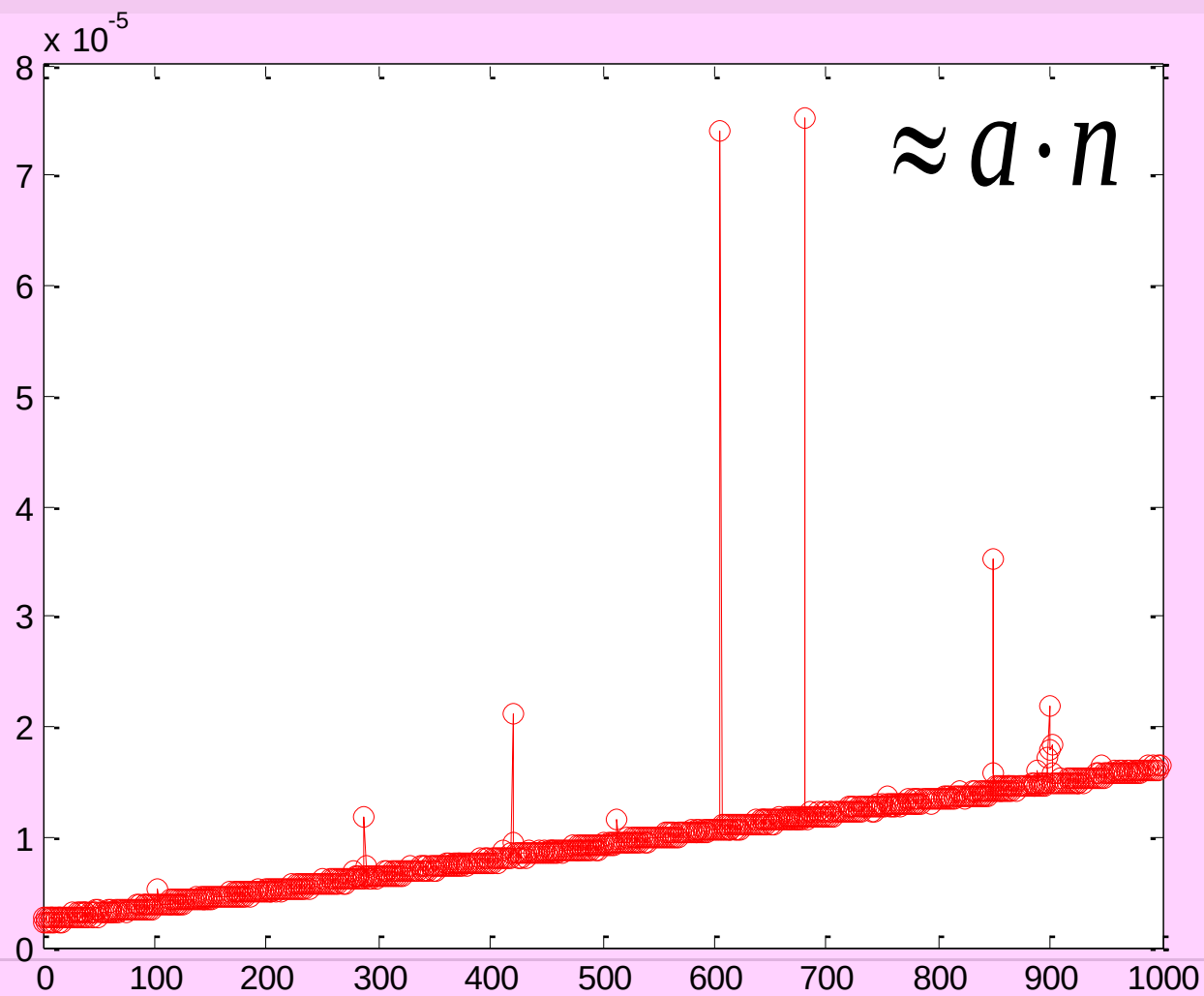
```
    end
```

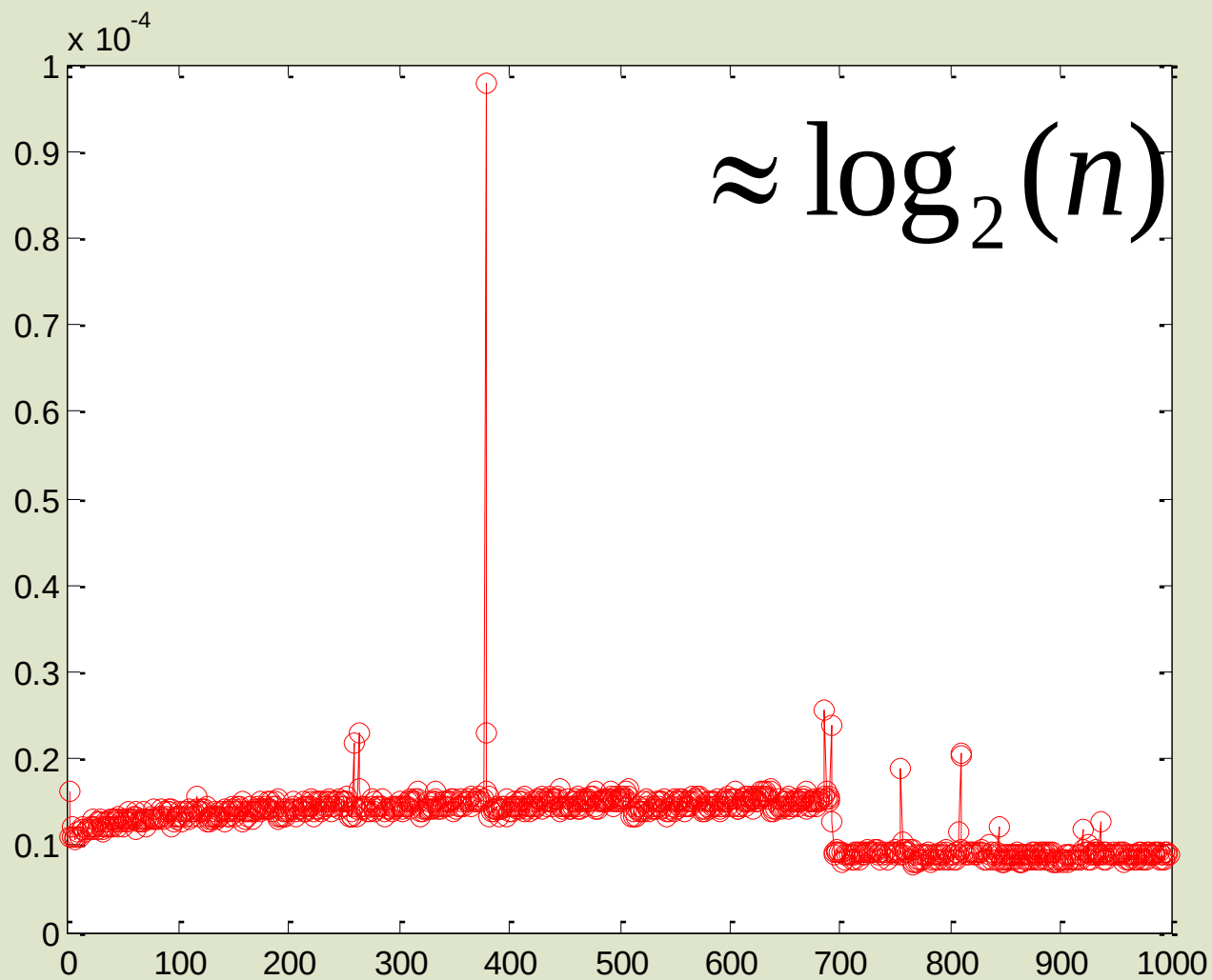
```
end
```


Γίνεται καλύτερα?

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}.$$

- ❑ `function fn = fibonacci3(n)`
 - ❑ `M = [1 1; 1 0];`
 - ❑ `F = M^n;`
 - ❑ `fn = F(1,2);`
-





Nature by numbers



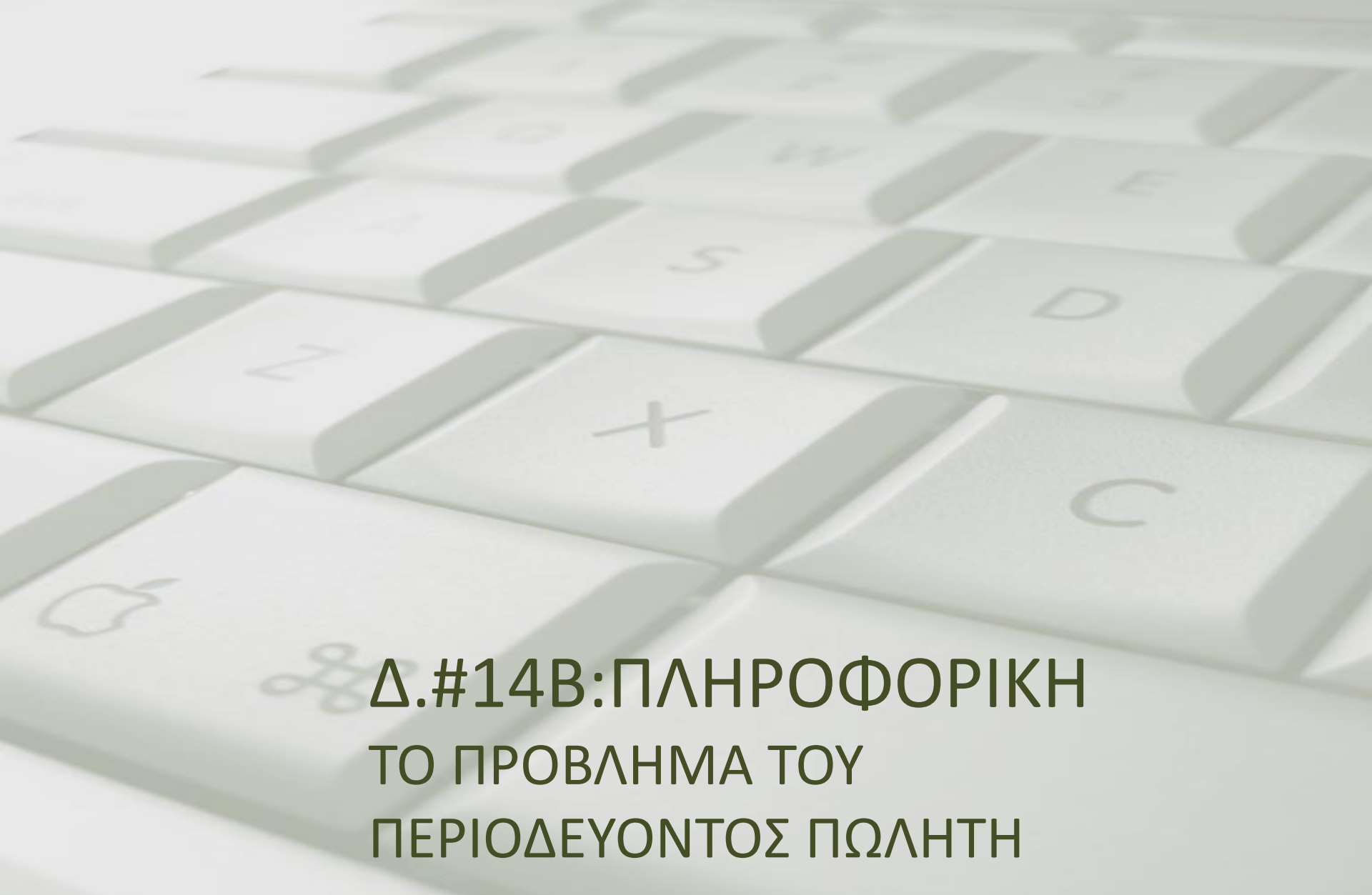
□ <http://vimeo.com/9953368>



Τέλος

Αν. Καθηγητής Κωνσταντίνος Καρατζάς

Τμήμα Μηχανολόγων Μηχανικών ΑΠΘ



Δ.#14B: ΠΛΗΡΟΦΟΡΙΚΗ ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΠΕΡΙΟΔΕΥΟΝΤΟΣ ΠΩΛΗΤΗ

21.04.2015

Αν. Καθ. Κωνσταντίνος Καρατζάς
Τμήμα Μηχανολόγων Μηχανικών ΑΠΘ

Περιεχόμενα



- ❑ Εισαγωγή στο πρόβλημα
 - ❑ Ορισμός
 - ❑ Χαρακτηριστικά
 - ❑ Μέθοδοι επίλυσης
 - ❑ Τρέχουσα κατάσταση
 - ❑ Εφαρμογές
-

Παρεπιπτόντως....

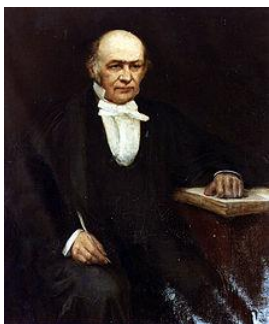
- Κια η εύρεση μίας ορίζουσας με τη μέθοδο των υποοριζουσών είναι πρόβλημα πολυπλοκότητας περιοδεύοντος πωλητή!
-

Εισαγωγή στο πρόβλημα

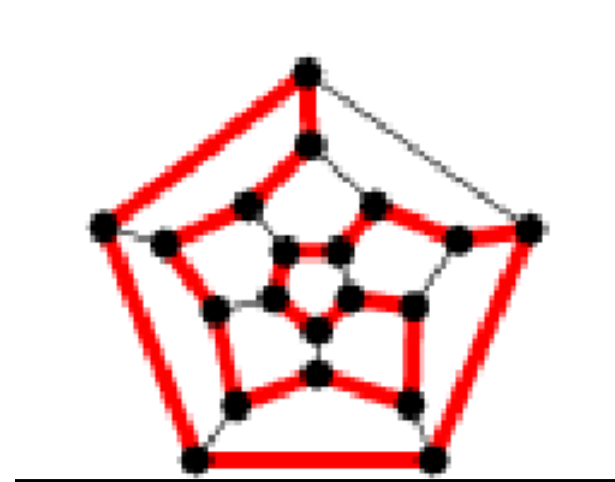
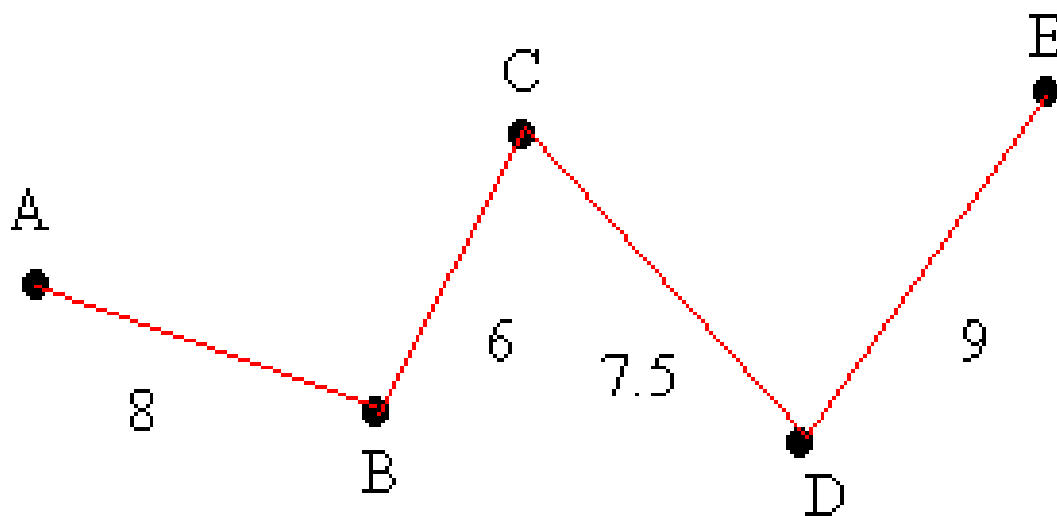


Ένας πωλητής θέλει να κάνει μία διαδρομή περνώντας μόνο μία φορά από ένα σύνολο πόλεων, και επιστρέφοντας στην αρχική. Ποιά διαδρομή αντιστοιχεί στη μικρότερη συνολική διανυόμενη απόσταση;

Ορισμοί



- Πλήρες γράφημα (μοναδική σύνδεση κορυφών)
- Γράφημα βαρών (ακμές με “κόστη”)
- Χαμιλτονιανή διαδρομή (πλήρες γράφημα βαρών με επιστροφή στην αρχή)
- Βέλτιστη Χαμιλτονιανή διαδρομή



Χαρακτηριστικά

- Επίκαιρο πρόβλημα με πολλές εφαρμογές
- Δοκιμή όλων των λύσεων απαραίτητη για εύρεση βέλτιστης ($O(n!)$)

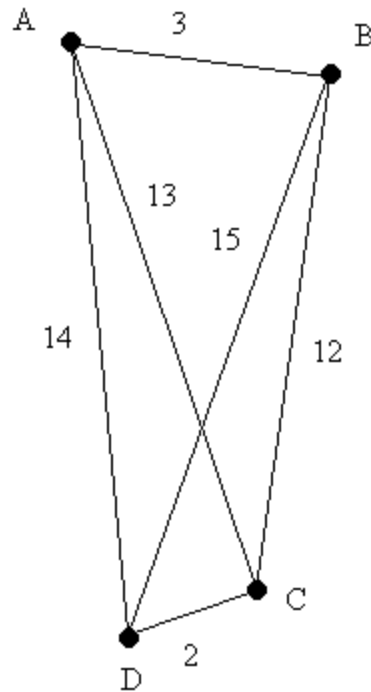
Μέθοδοι επίλυσης

- Αναλυτική
 - Βελτιστοποίησης
 - ▣ Είναι $O(n^2 2^n)$: βρίσκουν “βέλτιστες” λύσεις, απαιτείται αρκετός χρόνος
 - Προσεγγιστικές
 - ▣ Μέθοδος συνδέσμου χαμηλότερου κόστους
 - ▣ Μέθοδος εγγύτερου γείτονα
 - Ευρεστικές
 - ▣ Μέθοδος Χριστοφίδη
 - ▣ Νευρωνικά Δίκτυα
 - ▣ Νοημοσύνη Σμήνους
 - ▣ Νοημοσύνη αποικίας μυρμιγκιών
-

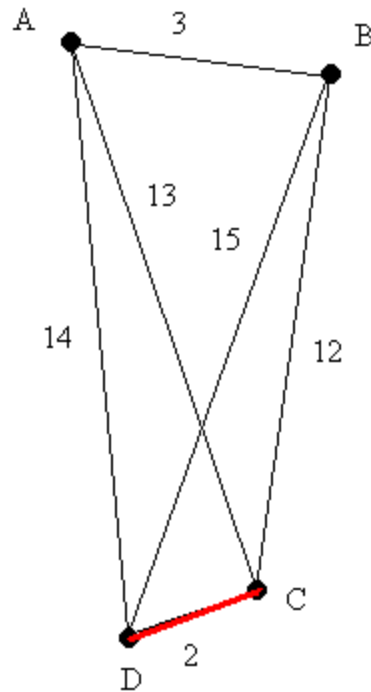
Μέθοδος χαμηλότερου κόστους

- Η ακμή με το χαμηλότερο κόστος επιλέγεται πρώτη
 - Ακολουθεί η ακμή με το αμέσως επόμενο χαμηλότερο κόστος
 - Συνεχίζω έτσι αρκεί να μην φτάσω σε περίπτωση σχηματισμού γραφήματος μικρότερου από το ζητούμενο, και εκτός εάν έχω κορυφή με τρεις τουλάχιστον ακμές
 - Περάτωση μόλις ολοκληρωθεί μία διαδριμή Χάμιλτον
-

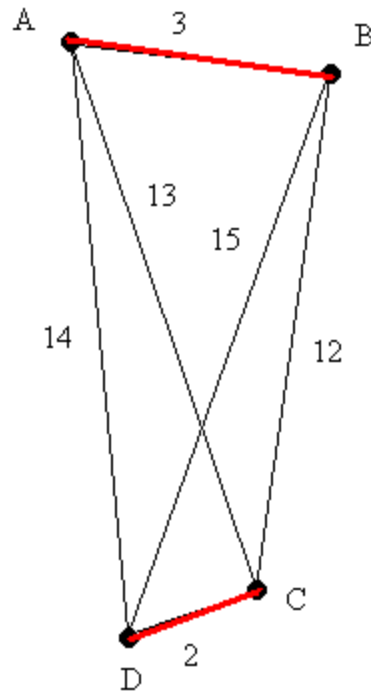
Μέθοδος χαμηλότερου κόστους



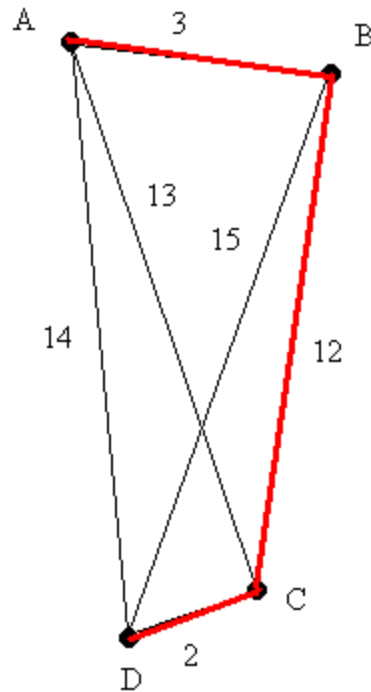
Μέθοδος χαμηλότερου κόστους



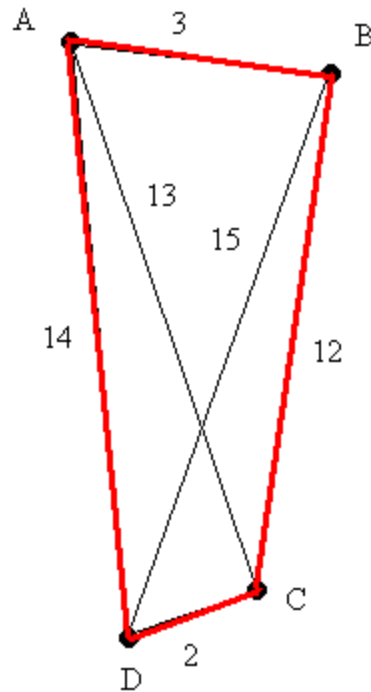
Μέθοδος χαμηλότερου κόστους



Μέθοδος χαμηλότερου κόστους



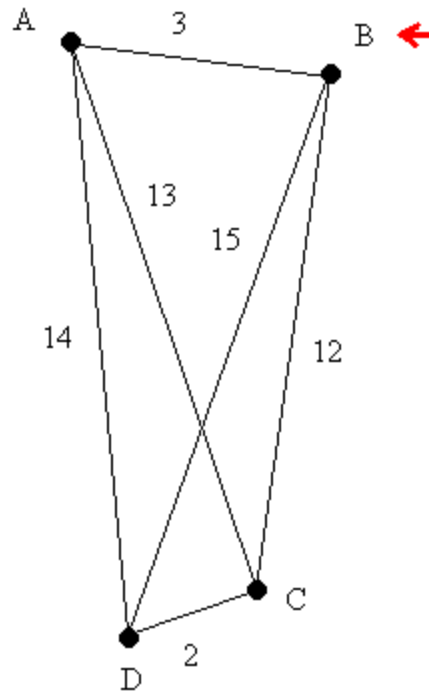
Μέθοδος χαμηλότερου κόστους



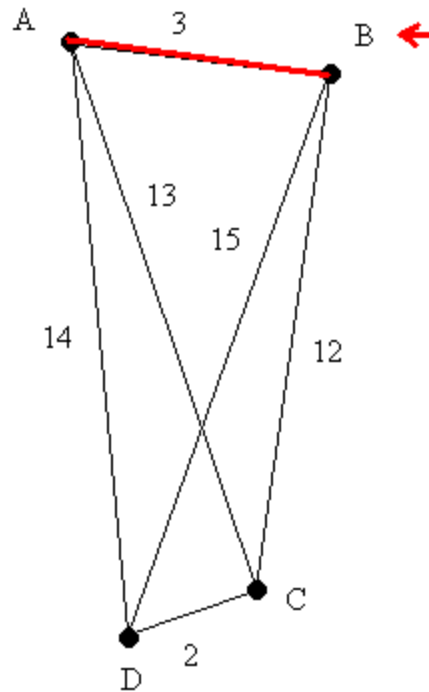
Μέθοδος εγγύτερου γείτονα

- Ξεκινώ από οποιαδήποτε κορυφή
 - Πηγαίνω στην εγγύτερη κορυφή που δεν έχω επισκευθεί
 - Συνεχίζω έως ότου έχω ολοκληρώσει διαδρομή Χάμιλτον
-

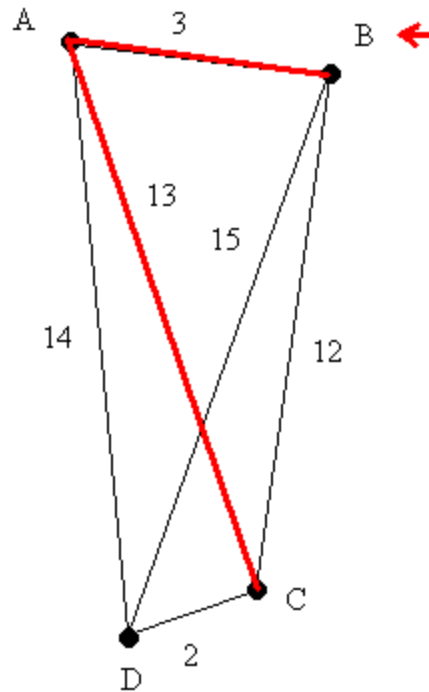
Μέθοδος εγγύτερου γείτονα



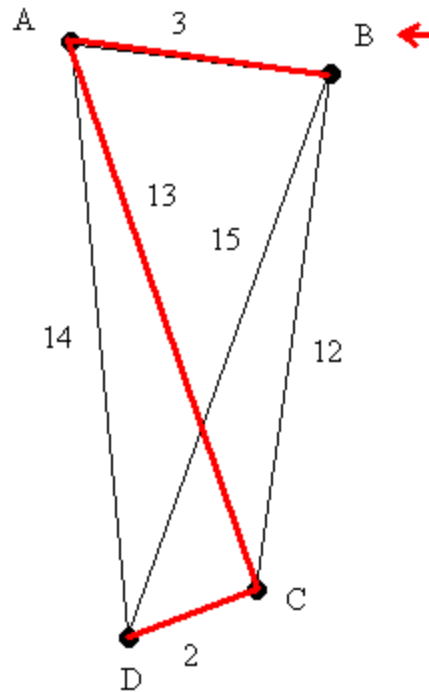
Μέθοδος εγγύτερου γείτονα



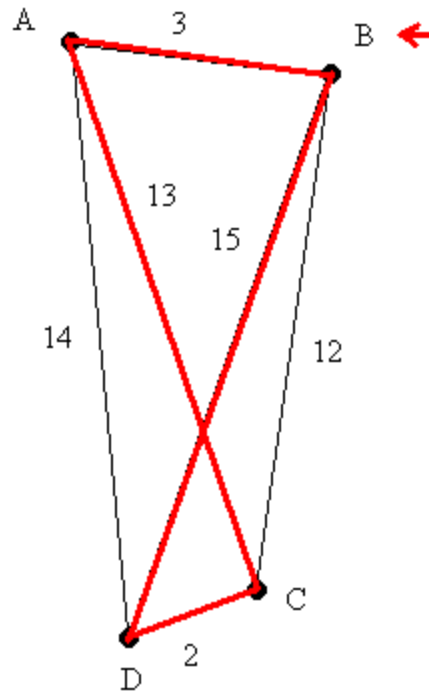
Μέθοδος εγγύτερου γείτονα



Μέθοδος εγγύτερου γείτονα

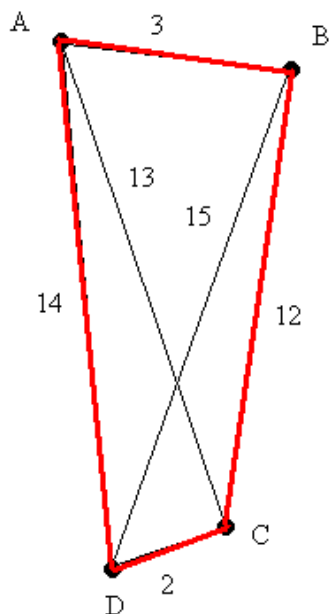


Μέθοδος εγγύτερου γείτονα



Σύγκριση μεθόδων

Χαμηλότερου κόστους

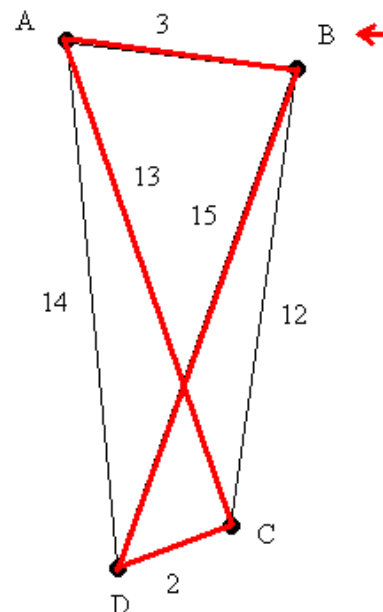


Συνολικό κόστος για:

Μέθ. χαμηλότερου κόστους – 31

Μέθ. εγγύτερου γείτ. – 33

Εγγύτερου γείτονα



Τρέχουσα κατάσταση

HELP! WE'RE LOST!

HELP "CAR 54"...AND WIN CASH
54...\$1,000 PRIZES
ONE...\$10,000 GRAND PRIZE

START-LINE FINISH

Map by Rand McNally

Help Toody and Muldoon find the shortest round trip route to visit all 33 locations shown on the map.
All you do is draw connecting straight lines from location to location to show the shortest round trip route.

HERE'S THE CORRECT START...
Begin at Chicago, Illinois. From there, lines show correct route as far as Erie, Pennsylvania. Next, do you go to Carlisle, Pennsylvania or Wana, West Virginia? Check the easy instructions on back of this entry blank for details.

© PROCTER & GAMBLE 1962

OFFICIAL RULES ON REVERSE SIDE

Τρέχουσα κατάσταση



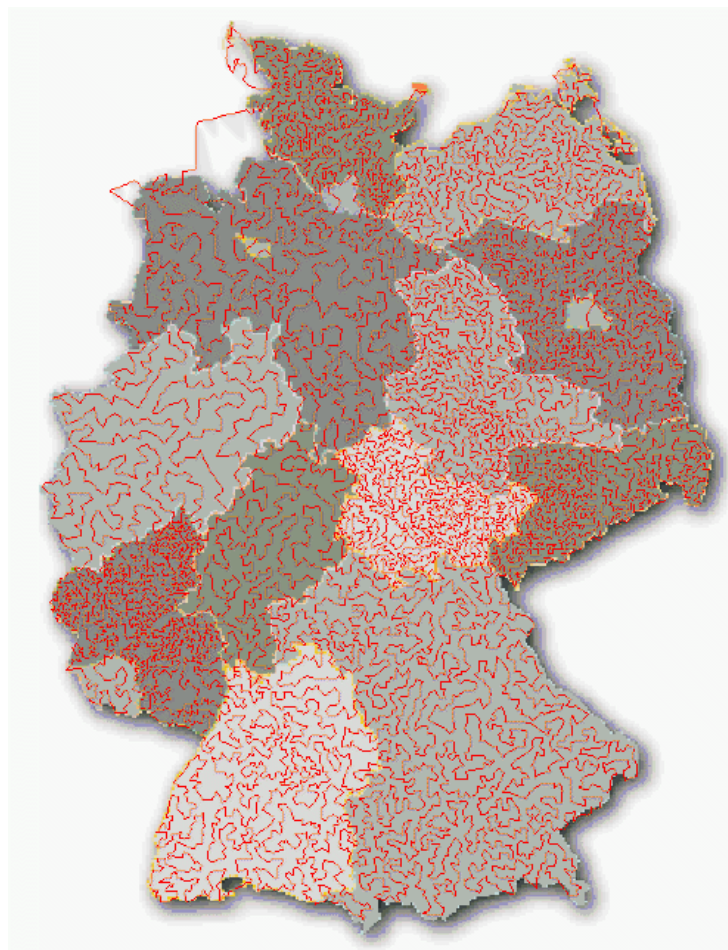
This tour of 13,500 US cities was generated by an advanced algorithm that used several “tricks” to limit the number of possible tours

<http://www.tsp.gatech.edu/>

*Required 5 “CPU-years”
(1998)*

Τρέχουσα κατάσταση

- Most recently: 15,112 cities in Germany
- April 21, 2001
- Approx 40,500 miles
- Used 110 different computers from Rice and Princeton University
- Total runtime: 22.6 years



Work in Progress – World Tour

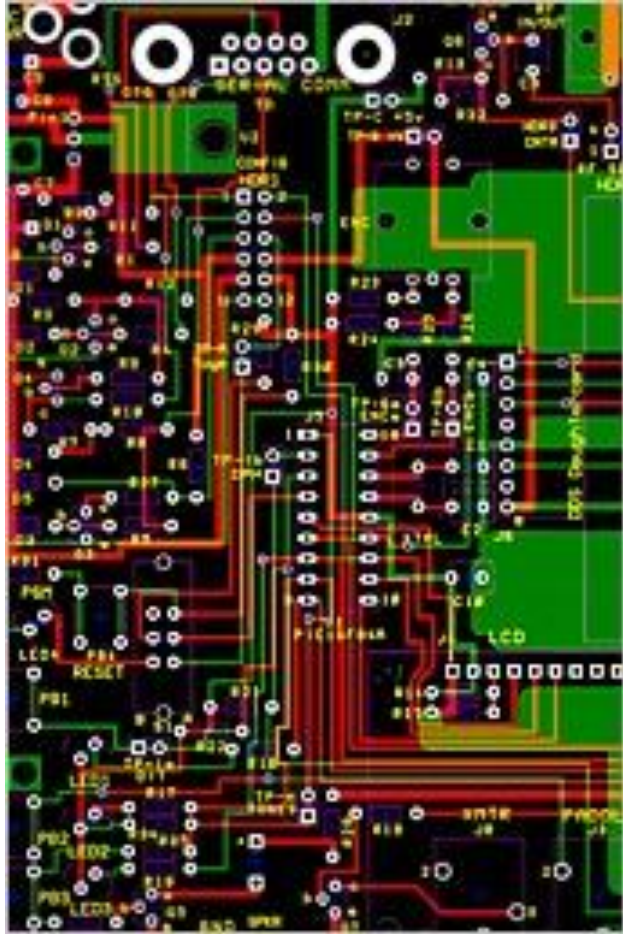
- 1,904,711 cities worldwide
- Keld Helsgaun
 - March 2004
 - .075% to .076% from estimated lower-bound



Εφαρμογές

- ▣ Εφοδιαστική – μεταφορές
 - Διαδρομές φορτηγών, σχολικών, delivery, κλπ..
 - ▣ CAM: Βιομηχανικό ρομπότ που επεξεργάζεται επιφάνειες
 - ▣ VLSI
 - ▣ Τηλεπικοινωνίες
 - ▣ Δρομολόγηση στο διαδίκτυο
 - ▣
-

Εφαρμογές

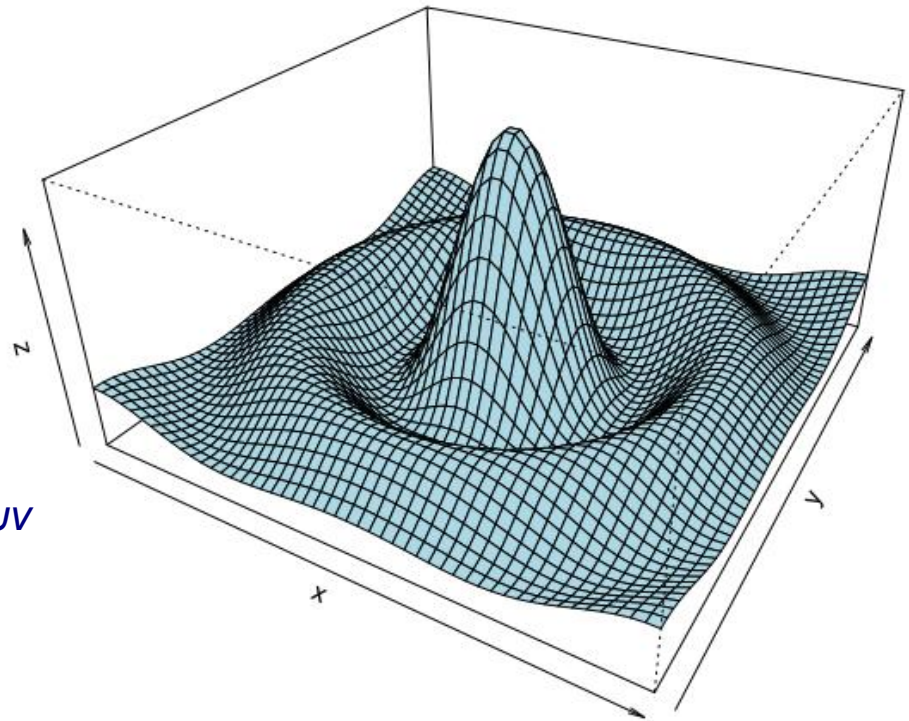


Βελτιστοποίηση

- Το ΠΠΠ είναι ένα από τα προβλήματα βελτιστοποίησης
- Στόχος: η εύρεση του κατάλληλου συνόλου παραμέτρων που μεγιστοποιούν (ή ελαχιστοποιούν) μία συνάρτηση

Γράφημα μίας συνάρτησης
 $z = f(x,y)$

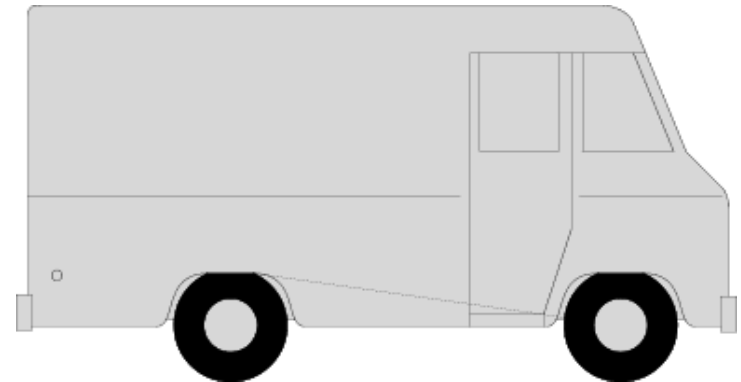
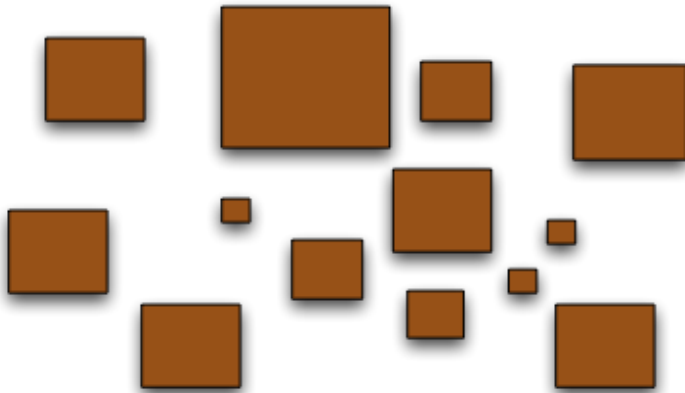
Ποιές τιμές των x και y δίνουν
την μεγαλύτερη τιμή της f ?



Παράδειγμα βελτιστοποίησης

□ Πόρτ μπαγκάζ

- Στόχος: επιλογή των δεμάτων που θα αφήσουν τον ελάχιστο ανεκμετάλλευστο χώρο

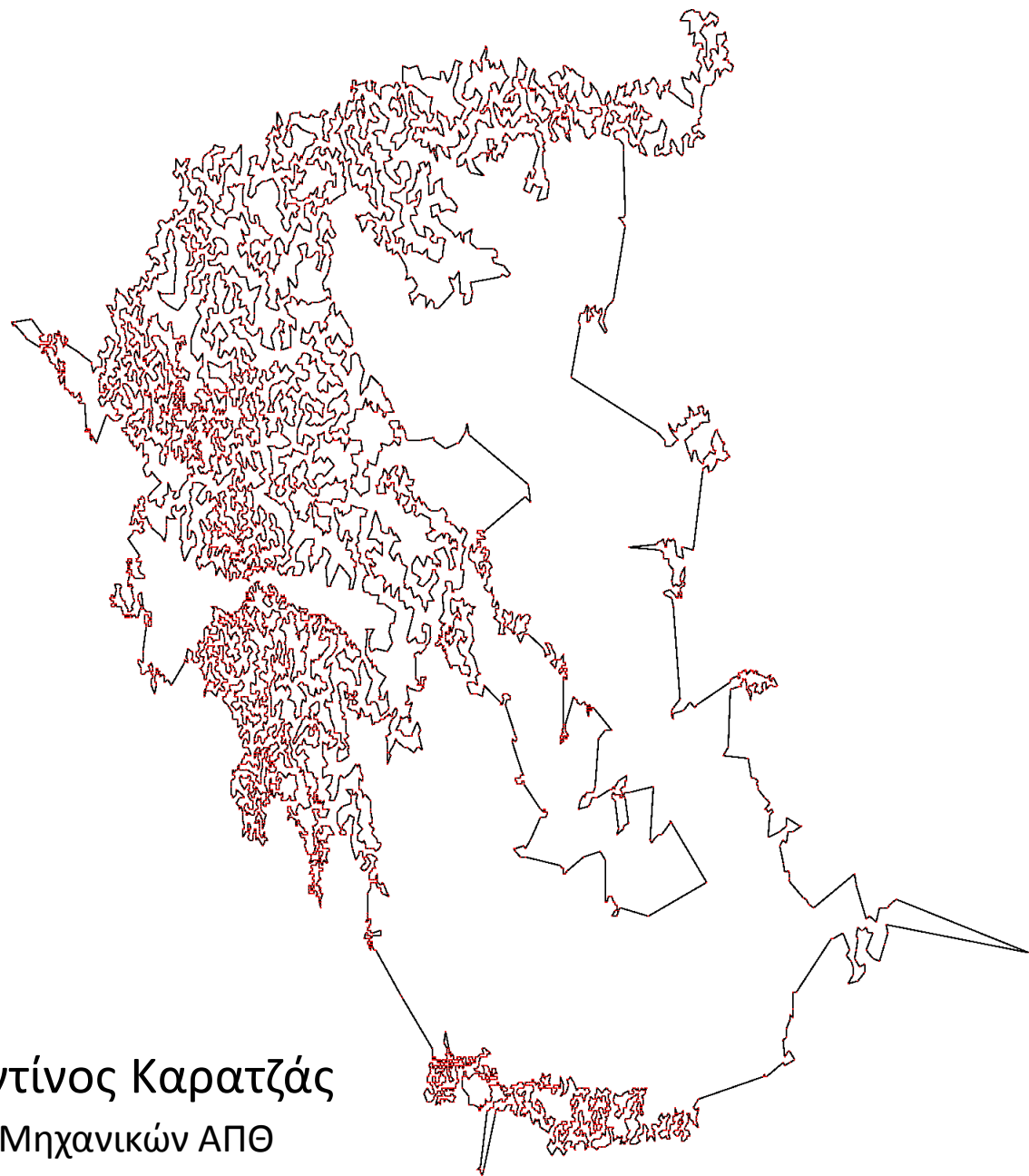


n boxes $\rightarrow 2^n$ combinations

The TSP Game

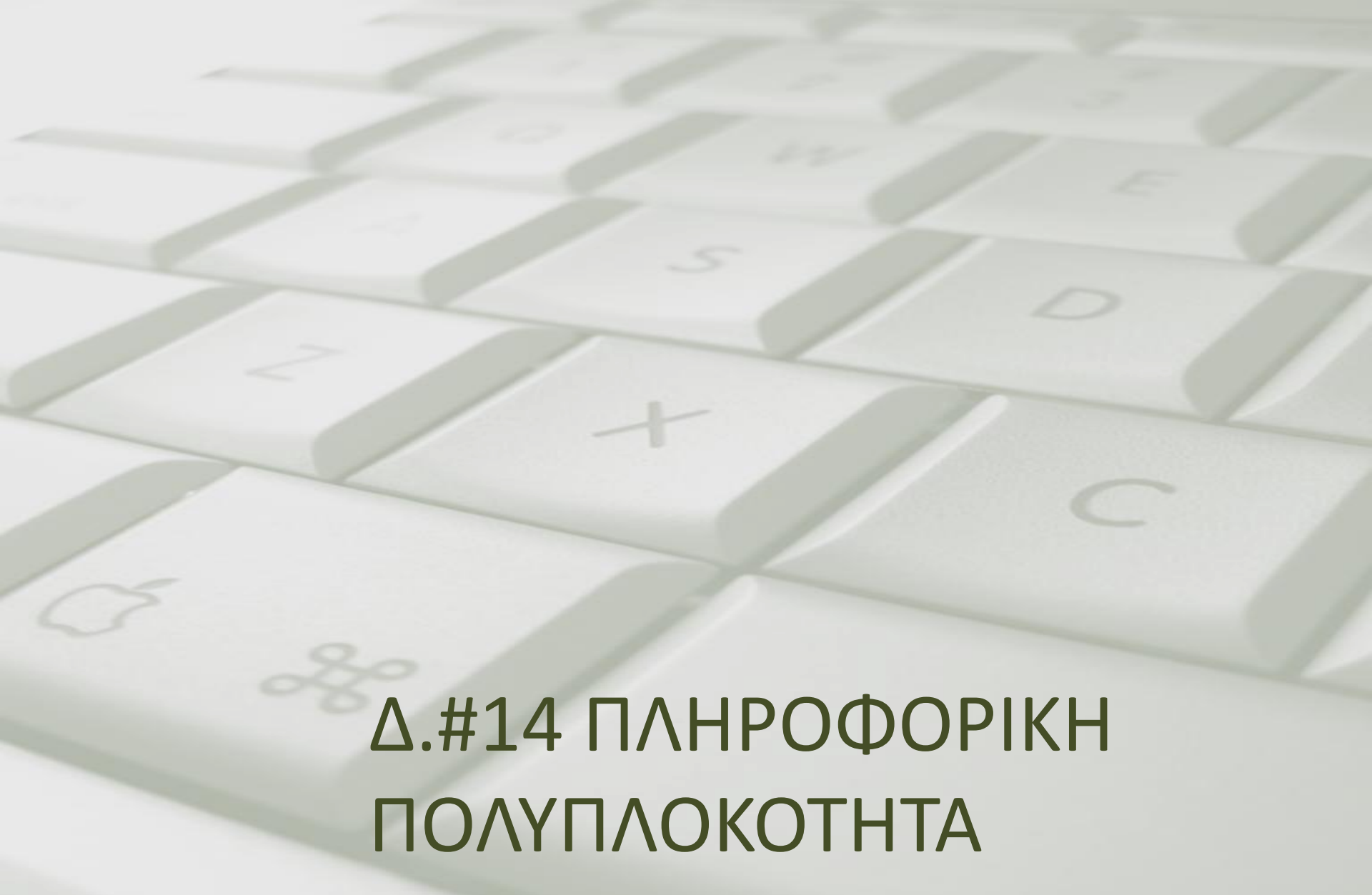
The screenshot shows a web browser window titled "TSP Games > Tour Finder" with the URL <http://www.tsp.gatech.edu/games/tspOnePlayer.html>. The browser's address bar also shows a search for "np complete travel". The page features a navigation menu on the left with links to Home, TSP Games, > Tour Finder (selected), and Tour Race. The main content area is titled "Tour Finder" and contains a description of the game: "The object of this game is to find the shortest possible **traveling salesman tour** through the green cities. The tour may start off at any of the cities, must touch upon each city at least once, and return to the originating city. It is fairly easy to find a good tour. Finding the best possible roundtrip is far less obvious." Below the text is a diagram showing a set of 7 green dots representing cities. A red line connects 6 of these dots in a sequence, forming a partial tour. The diagram is labeled "Cities: 7" and "Length: 574.00". At the bottom of the browser window, a status bar indicates "Applet tspOnePlayer started".

<http://www.tsp.gatech.edu/games/index.html>



Τέλος

Αν. Καθ. Κωνσταντίνος Καρατζάς
Τμήμα Μηχανολόγων Μηχανικών ΑΠΘ



Δ.#14 ΠΛΗΡΟΦΟΡΙΚΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑ

21.04.2015

Αναπλ. Καθηγητής Κωνσταντίνος Καρατζάς
Τμήμα Μηχανολόγων Μηχανικών ΑΠΘ

Υπολογιστική πολυπλοκότητα

- Fibonacci again
 - Γραμμικά συστήματα
 - Πίνακες-ορίζουσες
 - Μέθοδος Crammer
 - Αριθμός πράξεων
-

Fibonacci

$$f(n) = \begin{cases} 0, & n=0 \\ 1, & n=1 \\ f(n-1) + f(n-2), & n>1 \end{cases}$$



□ Πώς θα την υλοποιούσατε?

Fibonacci

```
function fn = fibonacci1(n)

if (n==0)
    fn = 0;
elseif (n==1)
    fn = 1;
else
    fn = fibonacci1(n-2)+fibonacci1(n-1);
end
```

Fibonacci

```
function fn = fibonacci2(n)
```

```
fn(1) = 1;
```

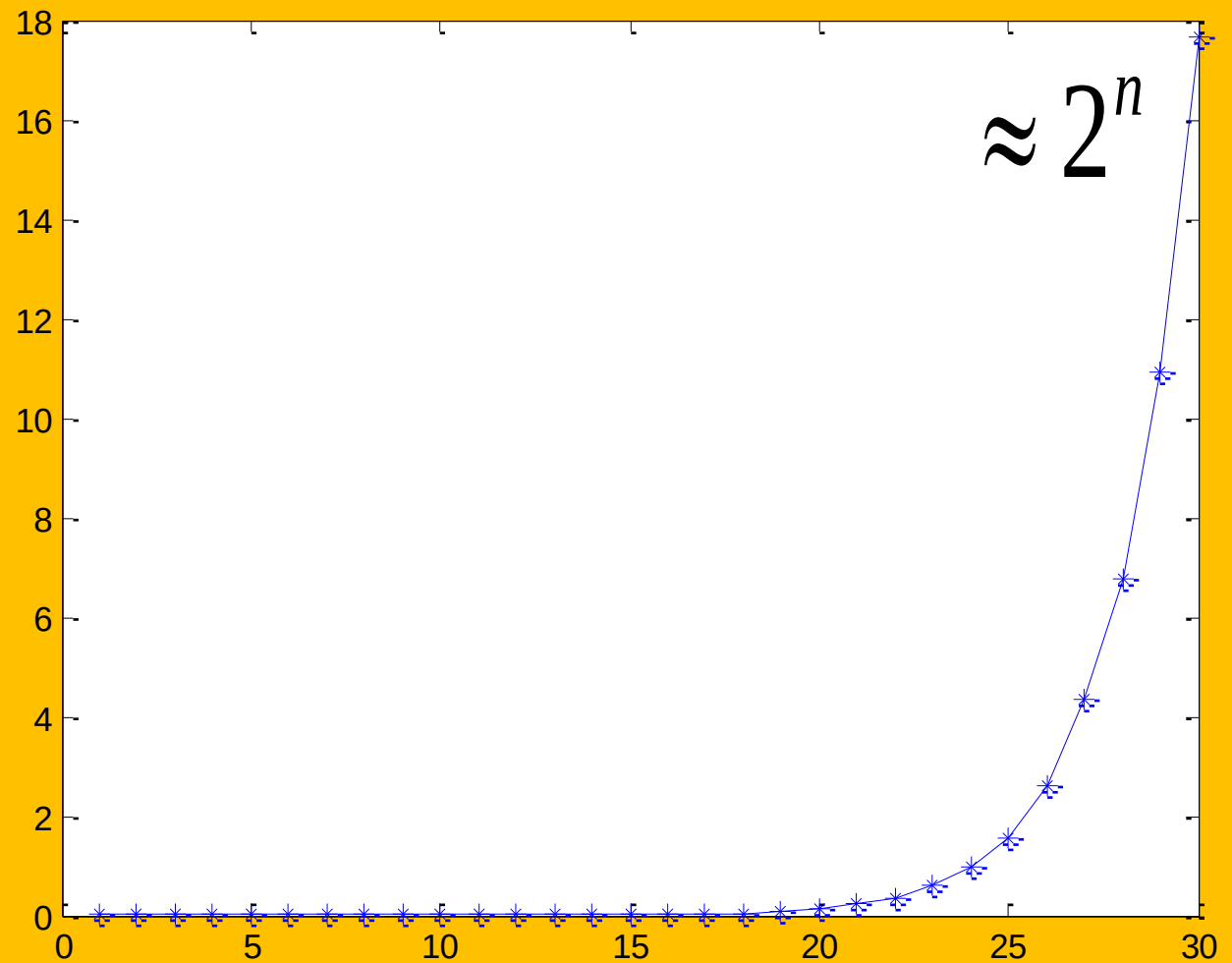
```
fn(2) = 1;
```

```
for i=3:n
```

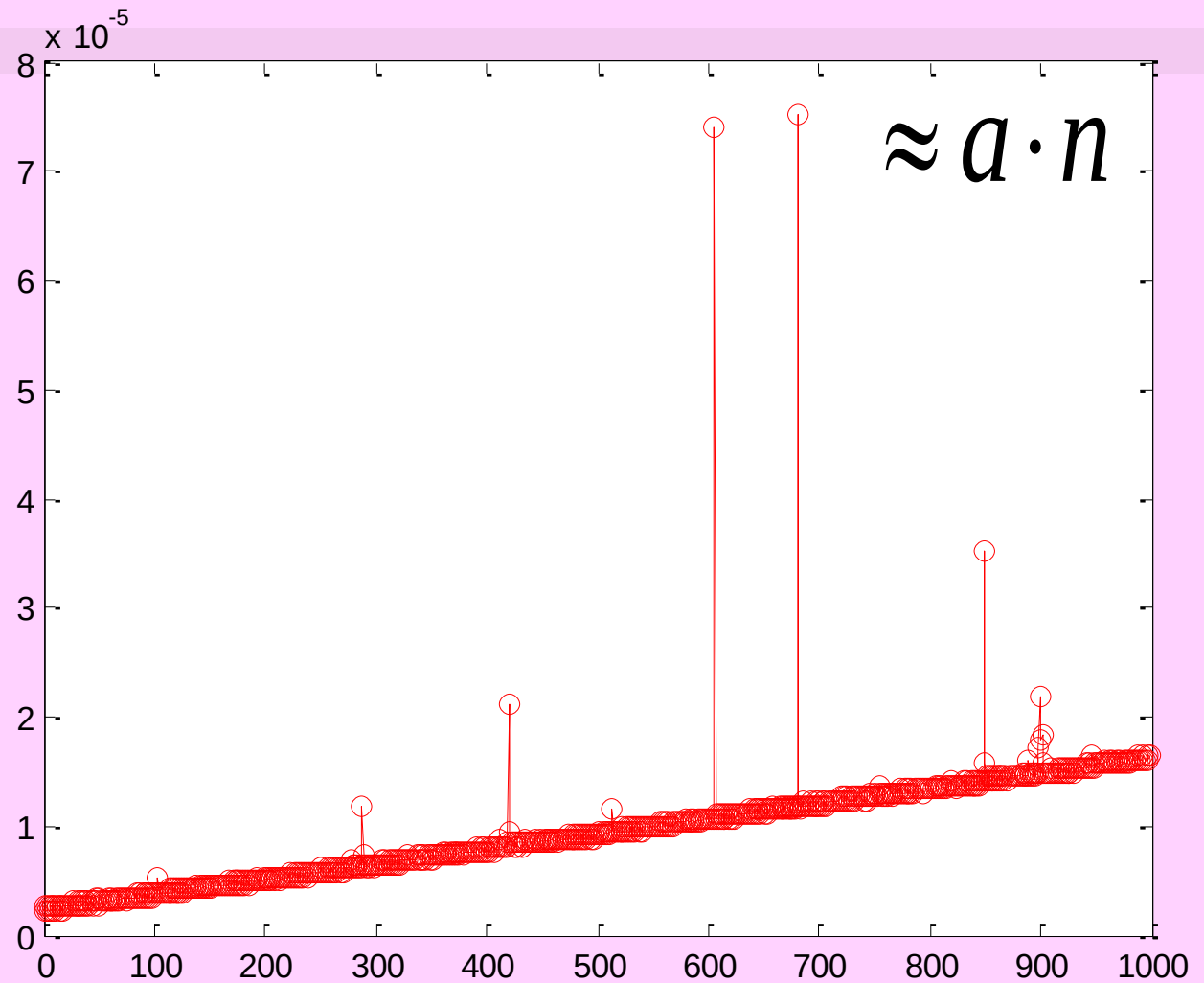
```
    fn(i) = fn(i-1) + fn(i-2);
```

```
end
```

Και πόσος χρόνος απαιτείται;



Και πόσος χρόνος απαιτείται;



Πώς προέκυψε αυτό?

```
function fn = fibonacci3(n)
```

```
    if (n==0)
```

```
        fn = 0;
```

```
    elseif (n==1)
```

```
        fn = 1;
```

```
    else
```

```
        f0 = 0;
```

```
        f1 = 1;
```

```
        for i=2:n
```

```
            fn = f1 + f0;
```

```
            f0 = f1;
```

```
            f1 = fn;
```

```
        end
```

```
end
```

Πώς προέκυψε αυτό?

```
function fn = fibonacci3(n)
```

```
if (n==0)
```

```
    fn = 0;
```

```
elseif (n==1)
```

```
    fn = 1;
```

```
else
```

```
    f0 = 0;
```

```
    f1 = 1;
```

```
    for i=2:n
```

```
        fn = f1 + f0;
```

```
        f0 = f1;
```

```
        f1 = fn;
```

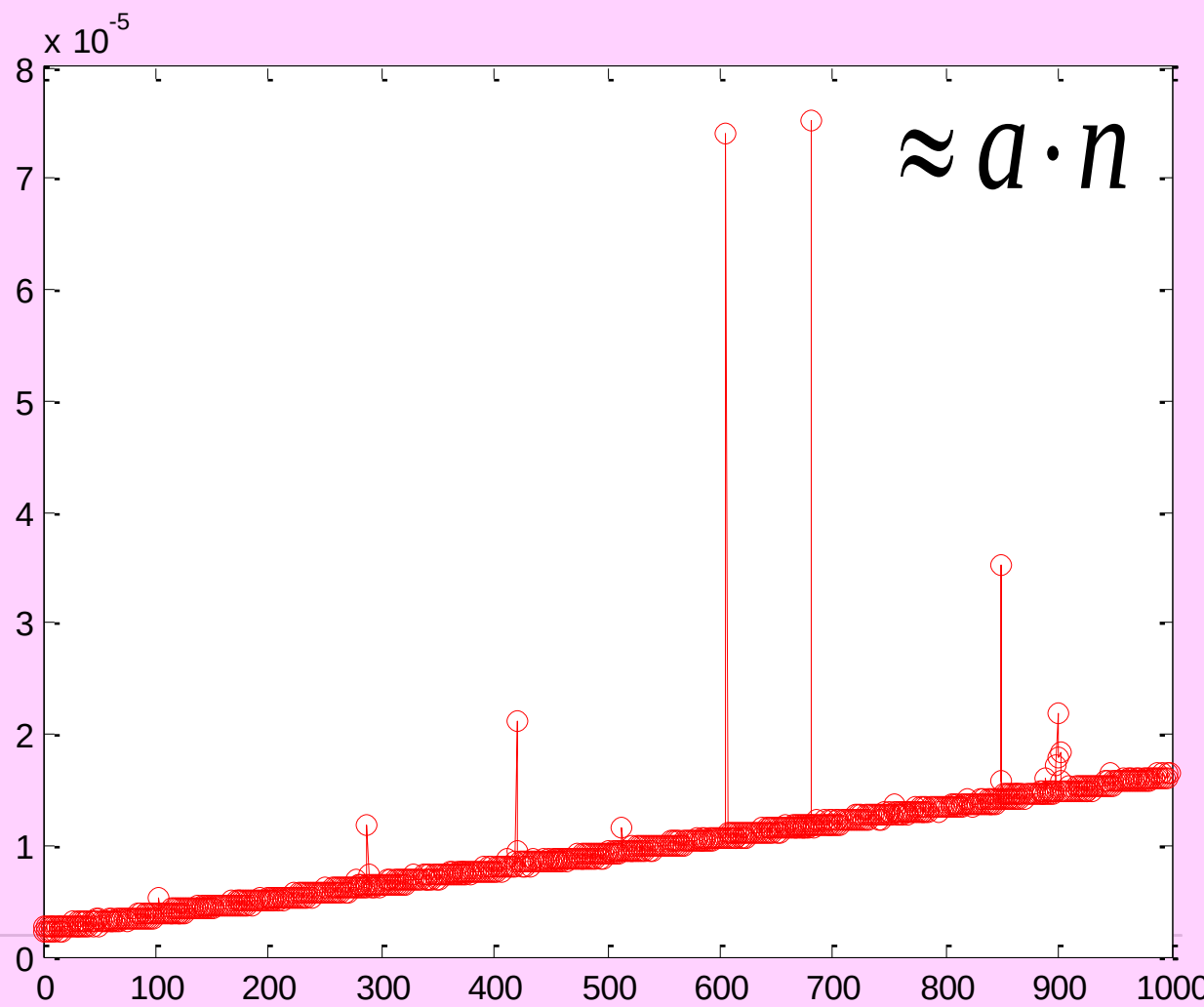
```
    end
```

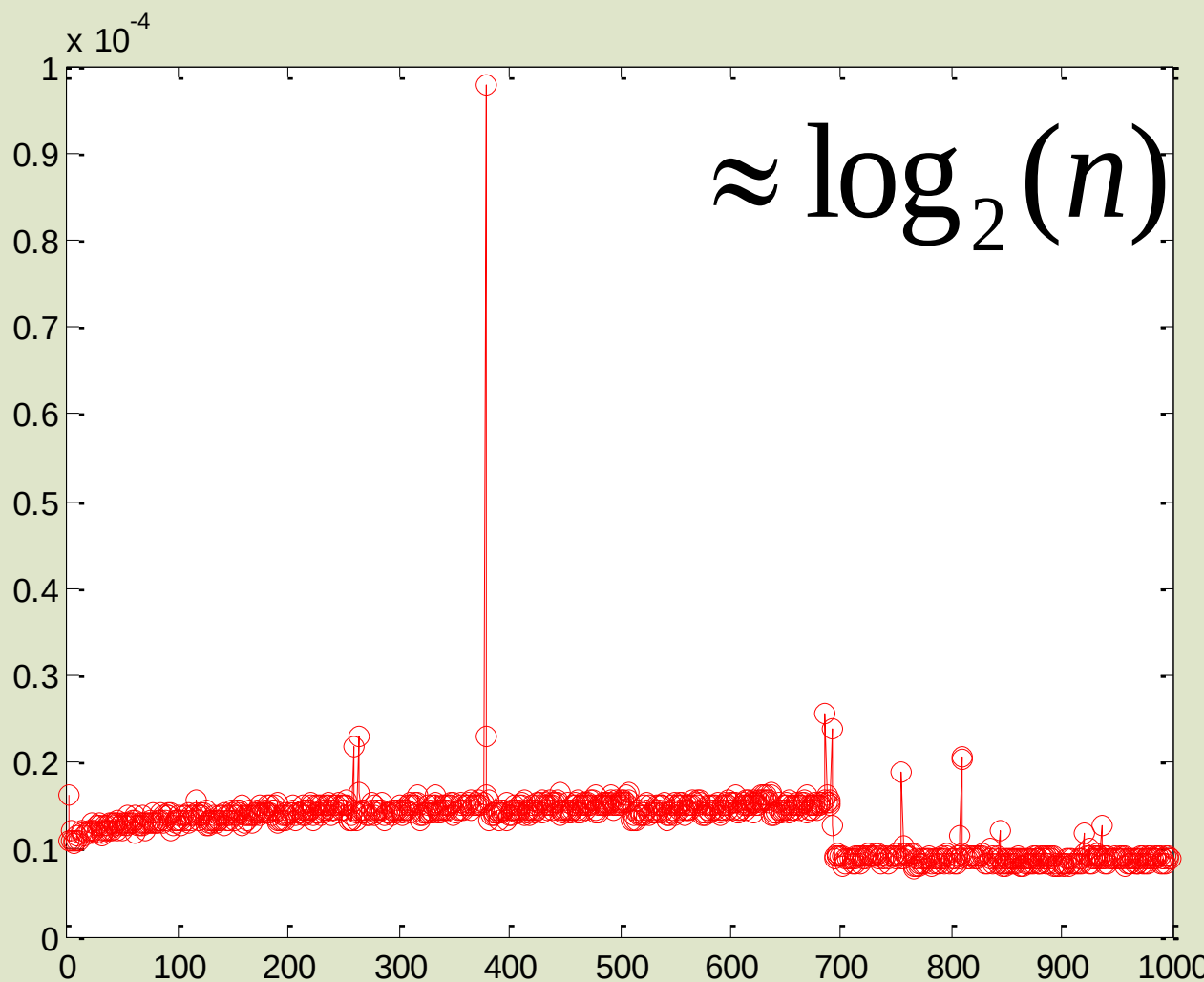
```
end
```

Γίνεται καλύτερα?

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}.$$

- ❑ `function fn = fibonacci4(n)`
 - ❑ `M = [1 1; 1 0];`
 - ❑ `F = M^n;`
 - ❑ `fn = F(1,2);`
-





αλγόριθμος Απαλοιφή Gauss

δεδομένα $n \times n$ πίνακας A , διάνυσμα $\mathbf{b}$

αποτελέσματα $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$

αρχή

για $i := 1$ έως $n-1$

για $k := i+1$ έως n

$m := a_{ki}/a_{ii}$

$a_{ki} := 0$

για $j := i+1$ έως n

$a_{kj} := a_{kj} - ma_{ij}$

τέλος για

τέλος για

τέλος για

για $i := n$ έως 1

για $j := i+1$ έως n

$b_i := b_i - a_{ij}x_j$

τέλος για

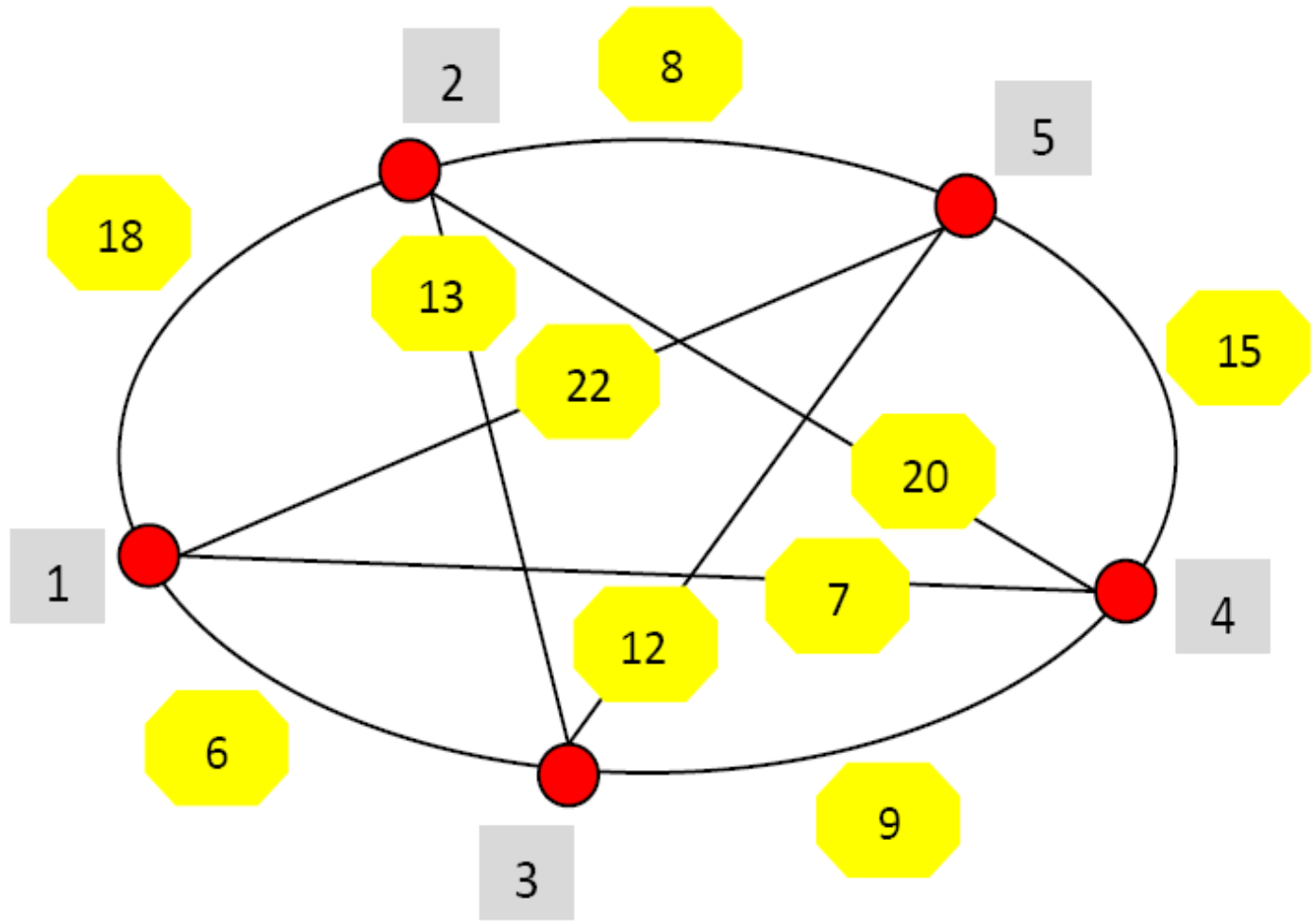
$x_i := b_i/a_{ii}$

τέλος για

τέλος

- Μέθοδος απαλοιφής Gauss
- Αριθμός στοιχειωδών πράξεων;

Το πρόβλημα του περιοδεύοντος πωλητή



Συναρτήσεις Υ.Π.

- Πολυπλοκότητα χρόνου & χώρου
 - Η έννοια της βασικής πράξης
-

Γραμμική συνάρτηση Π.Χ.

Παράδειγμα 1.1.1. Υπολογισμός της τετραγωνικής ρίζας

$$y = (\sum_{i=0}^n x_i^2)^{1/2}$$

αρχή

A:=0

για i:=1 **έως** n **κάνε**

A := A + x_i²

τέλος για

y:=sqrt (A)

τέλος

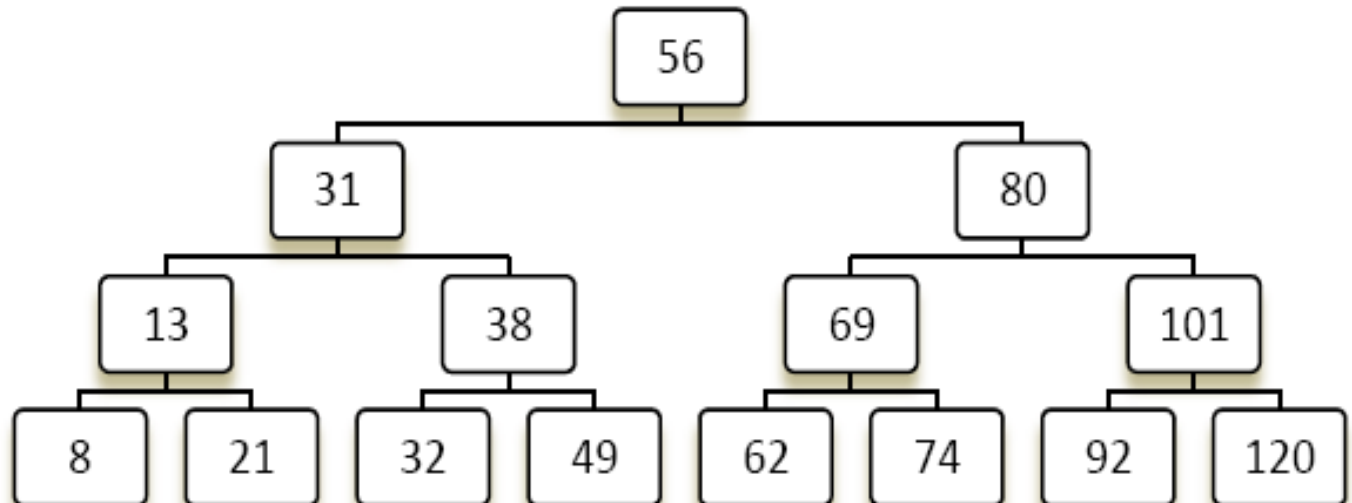
37

t(n)=n

Λογαριθμική

$K=[8,13,21,31,32,38,49,56,62,69,74,80,92,101,120]$

Περιλαμβάνει τον αρ. 32;



δεδομένα πίνακας $K = (k_1, k_2, \dots, k_n)$

αποτελέσματα τιμή λογικής μεταβλητής found
αρχή

found:=false

first:=1; last:=n

όσο (first≤last) και (found:=false) κάνε

$m := \left\lfloor \frac{last + first}{2} \right\rfloor$

αν $x := K[m]$

τότε found:=true

αλλιώς αν $x < K[m]$

τότε last:=m-1

αλλιώς first:=m+1

τέλος αν

τέλος αν

τέλος όσο

αν found:=true

τότε index:=m {επιτυχία, με $K[index]=x$ }

αλλιώς index:=0 {αναζήτηση ανεπιτυχής}

τέλος αν

τέλος

Λογαριθμική συνάρτηση Π.Χ.

$$K = (k_1, k_2, \dots, k_n)$$

Διαδική αναζήτηση

$$t(n) = \log_2(n)$$

$n \log_2 n$

Παράδειγμα: ταξινόμηση με επιλογή

- Έστω αταξινόμητος πίνακας $A = (\alpha_1, \alpha_2, \dots, \alpha_n)$
 - Ταξινομήστε σε μή φθίνουσα σειρά
-

Αλγόριθμος ταξινόμηση με επιλογή

δεδομένα αταξιινόμητος πίνακας $A = (a_1, a_2, \dots, a_n)$

αποτελέσματα ταξινομημένος πίνακας $A = (a_1, a_2, \dots, a_n)$

αρχή

για $i := 1$ έως n

$A[k] := \min\{A[i], A[i+1], \dots, A[n]\}$ (θεωρούμε ότι

τα στοιχεία από 1 έως $A[i-1]$ έχουν ήδη ταξινομηθεί

αντιμετάθεσε $A[i]$ με $A[k]$

τέλος για

τέλος

$$t(n) = n \log_2 n$$

Κυβική συνάρτηση Π.Χ.

αρχή

για $i:=1$ έως n κάνε

για $j:=1$ έως n κάνε

$S:=0$

για $k:=1$ έως n κάνε

$S:=S+A[I, j]B[I, j]$

τέλος για

$C[I, j]:=S$

τέλος για

τέλος για

τέλος

$$t(n)=n^3$$

Εκθετική συνάρτηση Π.Χ.

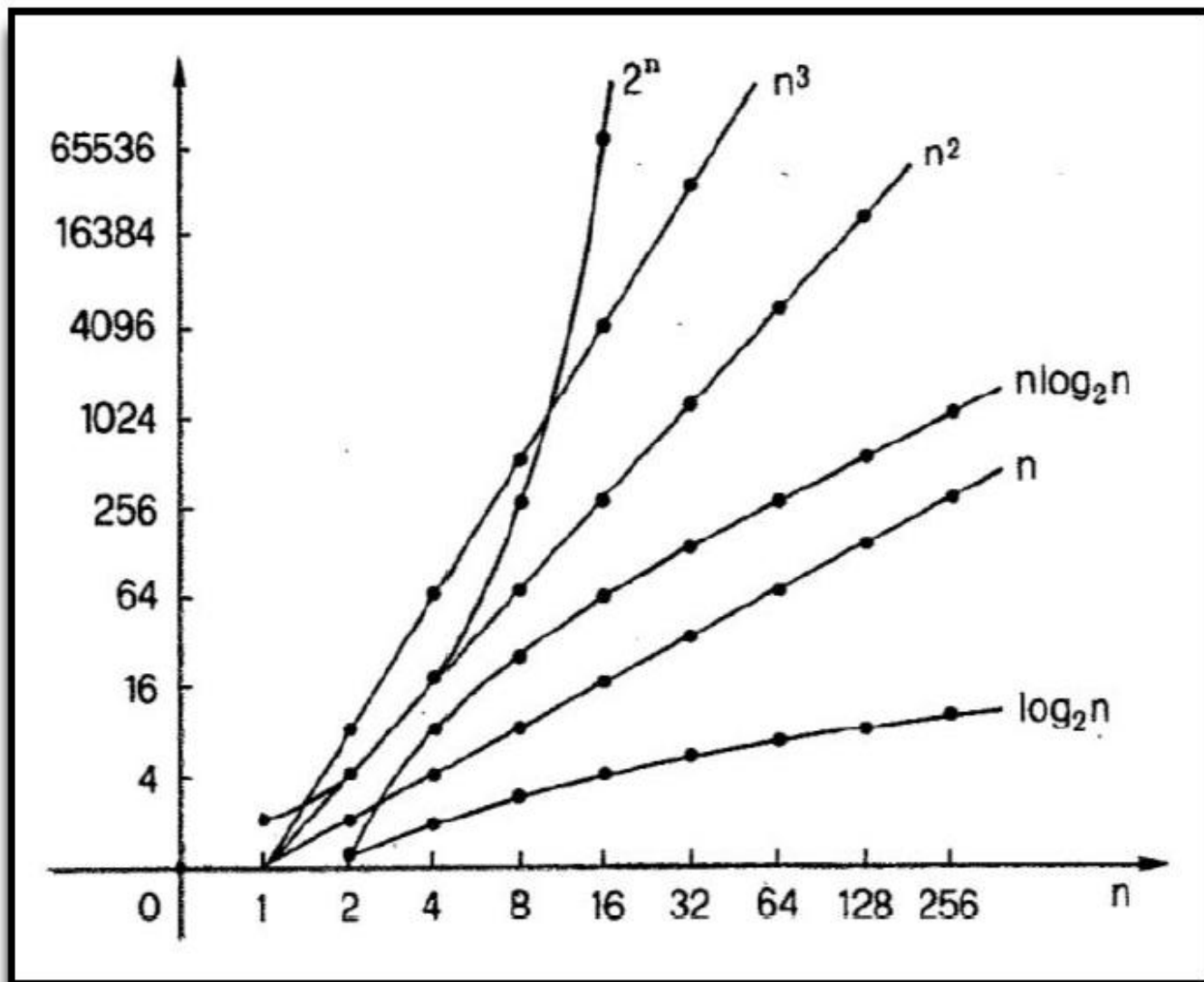
Παράδειγμα: έστω σύνολο S , n αριθμών

- Υπάρχει υποσύνολο του S , τ.ώ. το άθροισμα των στοιχείων του να ισούται με δεδομένο αριθμό a ;
- Τα δυνατά υποσύνολα είναι 2^n .

$$t(n) = 2^n$$

Συγκρίνοντας σ.π.

n	$\log_2 n$	$n \log_2 n$	n^2	n^3	2^n
2	1	2	4	8	4
10	3.322	33.22	100	10^3	$>10^3$
100	6.664	664.4	10000	10^6	$>10^{25}$
1000	9.996	9966.0	10^6	10^9	$>10^{250}$
10000	13.287	132877.0	10^8	10^{12}	$>10^{2500}$



Πολυπλοκότητα καλύτερης, μέσης & χειρίστης περίπτωσης

- Είναι ο αλγόριθμος έρμαιο του προβλήματος;;;
 - Παράδειγμα: ακολουθιακή αναζήτηση
-

Ασυμπτωτική πολυπλοκότητα

- Επιστήμη Μηχανικού => «μεγάλα» προβλήματα
- Ανάλυση πολυπλοκότητας: «τελική» συμπεριφορά αλγορίθμων
- Παράδειγμα: για το ίδιο πρόβλημα, $t_1(n)=100n$ και $t_2(n)=0.1n^2$.

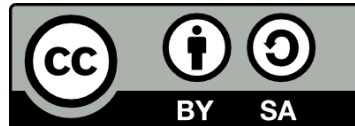
Σημείωμα Αναφοράς

- Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Κωνσταντίνος Καρατζάς. «Πληροφορική. Ενότητα 7: Α. Πληροφορική: Ακολουθία Fibonacci- Αναδρομικότητα- Πολυπλοκότητα. Β. Το πρόβλημα του περιοδεύοντος πωλητή». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://opencourses.auth.gr/courses/OCRS328/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Παρόμοια Διανομή [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

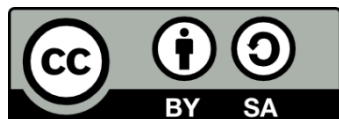
[1] <http://creativecommons.org/licenses/by-sa/4.0/>





Τέλος ενότητας

Θεσσαλονίκη, Εαρινό Εξάμηνο 2014-2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Σημειώματα

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

