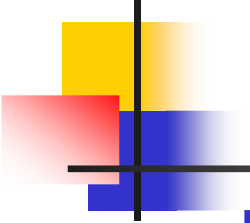


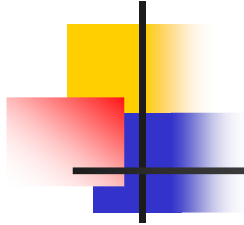


## Δομές Δεδομένων 352

---

Εγγραφή στο eclass (Πρέπει να έχετε  
λογαριασμό στο eclass).  
Το μάθημα είναι το DI281

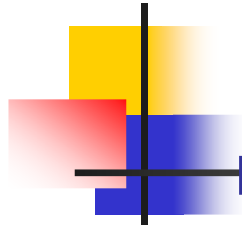
- 
- **Data Structures & Algorithms in JAVA M.T. Goodrich & R.Tamassia 5<sup>th</sup> Edition, Εκδόσεις Δίαυλος 2013**
  - **Αλγόριθμοι σε Java Μέρη 1-4, R.Sedgewick, Εκδόσεις Κλειδάριθμος**
  - Δομές Δεδομένων, Γ.Γεωργακόπουλος, Πανεπιστημιακές Εκδόσεις Κρήτης
  - Δομές Δεδομένων, Αλγόριθμοι, και Εφαρμογές στη C++, S.Sahni, Εκδόσεις Τζιόλα
  - Δομές Δεδομένων, Μισιρλής



## Απαιτήσεις για να περάσω το μάθημα

---

1. Θα έχετε δύο ασκήσεις προγραμματιστικές. Αυτές συμμετέχουν στο 40% της διαμόρφωσης του τελικού βαθμού.
2. Το τελικό γραπτό διαγώνισμα συμμετέχει κατά 60% στη διαμόρφωση του τελικού βαθμού.
3. Θα πρέπει όμως και στα δύο να έχετε προβιβάσιμο βαθμό.
4. Για να μπορέσετε να παρακολουθήσετε το μάθημα (αλλά και τις ασκήσεις) θα πρέπει να ξέρετε προγραμματισμό JAVA




Μεγάλο μέρος της ύλης είναι στα βιβλία και θα συμπληρωθεί με το περιεχόμενο των μαθημάτων.

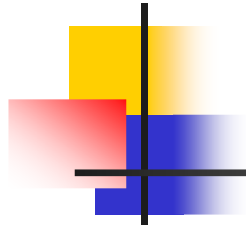
Είναι καλό να συμμετέχετε και να μην αφήσετε τα πάντα για την περίοδο των εξετάσεων. Τότε τα πράγματα ίσως είναι δύσκολα.

Κατά την διάρκεια του μαθήματος αν υπάρχουν απορίες μπορείτε να διακόπτετε και να ρωτάτε. Μην φοβάστε να ρωτήσετε, να σκέφτεστε ότι αν δεν καταλάβετε κάτι δεν φταίτε εσείς αλλά εγώ που δεν το εξήγησα καλά.  
Όταν διαβάζετε αν έχετε απορίες το mail μου είναι [mike@di.uoa.gr](mailto:mike@di.uoa.gr)

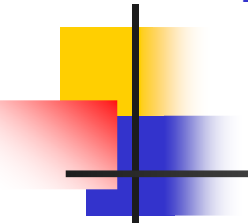
Δεν σας κάνω χάρη είμαι υποχρεωμένος να σας απαντήσω.

# Πληροφορική

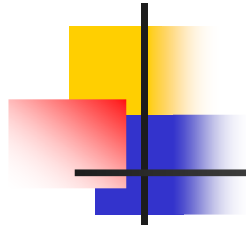
- 
- Μαθηματικά (ανακαλύπτουν σημαντικά γεγονότα και συσχετίσεις που μπορούν να εκφρασθούν με συμβολική ή ποσοτική γλώσσα)
  - Φυσικές Επιστήμες (προσπαθεί να ανακαλύψει κανόνες και αρχές της φύσης)
  - Engineering (σχεδιασμός και κατασκευή προϊόντων)



Η πληροφορική είναι μια επιστήμη που αφορά τεχνητά πράγματα –αυτά που κατασκευάζουν ανθρώπινα όντα- αντί για πράγματα που συμβαίνουν στη φύση που δεν έγιναν από ανθρώπινα όντα. Αντί για φυσική επιστήμη είναι η *επιστήμη του τεχνητού*.

- 
- Η ανάπτυξη υπολογιστικών συστημάτων ή ο σχεδιασμός αλγορίθμων είναι ουσιαστικά engineering δραστηριότητες.
  - Η πειραματική συλλογή δεδομένων και η μελέτη της συμπεριφοράς αντικειμένων που έχουν δημιουργηθεί με αυτόν τον τρόπο είναι ουσιαστικά επιστημονική δραστηριότητα.
  - Η ανακάλυψη και ανάπτυξη τυπικών θεωριών που εκφράζουν ουσιαστικές ιδέες που υποκρύπτονται κάτω από μια κλάση υπολογισμών είναι μαθηματικές δραστηριότητες.

## Επομένως πρέπει:

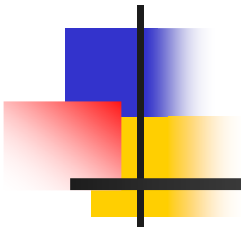


- Ικανότητα στη μαθηματική ανάλυση και την μαθηματική αιτιολόγηση
- Κατανόηση των πειραματικών επιστημονικών μεθόδων και εκτέλεσης πειραμάτων, συλλογής δεδομένων και ανακάλυψης σημασίας.
- Κατανόηση του πως λειτουργούν τα πράγματα, σχεδιασμό αλγορίθμων, επιλογή σωστών δομών δεδομένων και ανάπτυξη λογισμικού για επίλυση προβλημάτων



# Εισαγωγή (1)

- **Αλγόριθμος:** Μια βήμα-βήμα διαδικασία για την εκτέλεση μιας εργασίας από ένα υπολογιστή.
- **Δομή Δεδομένων:** Ένας συστηματικός τρόπος οργάνωσης και προσπέλασης στα δεδομένα.
  - Οι αλγόριθμοι και οι δομές δεδομένων αποτελούν σημαντικό στοιχείο των συστημάτων λογισμικού
  - Οι καλύτεροι αλγόριθμοι οδηγούν σε καλύτερα προγράμματα
  - Θα μάθουμε πως να σχεδιάζουμε, να αναλύουμε και να υλοποιούμε βασικούς αλγόριθμους και δομές δεδομένων.



## Εισαγωγή (2)

Οι Δομές Δεδομένων πραγματεύονται την ταχεία αλγοριθμική επίλυση μιας κατηγορίας προβλημάτων εντός του διαθέσιμου χώρου.

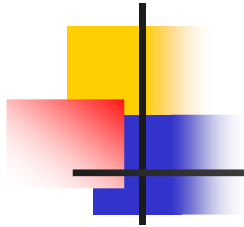
Ονομάζουμε δεδομένο την συμβολική παράσταση κάθε πληροφορίας.

Ορίζουμε σαν ζητούμενο την συμβολική παράσταση μιας επιθυμητής αλλά μη διαθέσιμης πληροφορίας

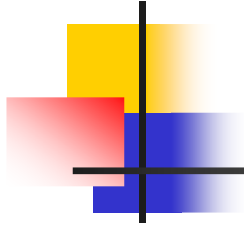
Πρόβλημα: κάθε σαφώς ορισμένη σχέση ανάμεσα σε δεδομένα και ζητούμενα.

## Εισαγωγή (3)

### Τύπος προβλημάτων που αντιμετωπίζουμε



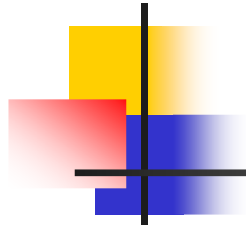
- Έχουμε δυναμικό σύνολο δεδομένων, δηλαδή ένα σύνολο δεδομένων το οποίο μεταβάλλεται με εισαγωγές και διαγραφές
- Στα δεδομένα μας θέλουμε να εκτελέσουμε συγκεκριμένες λειτουργίες
- Επίλυση του προβλήματος: κάθε υπολογισμό που από μια συγκεκριμένη περιγραφή της δομής κατασκευάζει τα ζητούμενα ακολουθώντας μια σειρά επιτρεπτών πράξεων με βάση σταθερές οδηγίες (αλγόριθμος)



## Θέματα

---

- Ανάλυση Αλγορίθμων
- Βασικές Δομές Δεδομένων
- Αναζήτηση
- Ταξινόμηση
- Γράφοι
- Επεξεργασία Κειμένου?



# Ανάλυση Αλγορίθμων

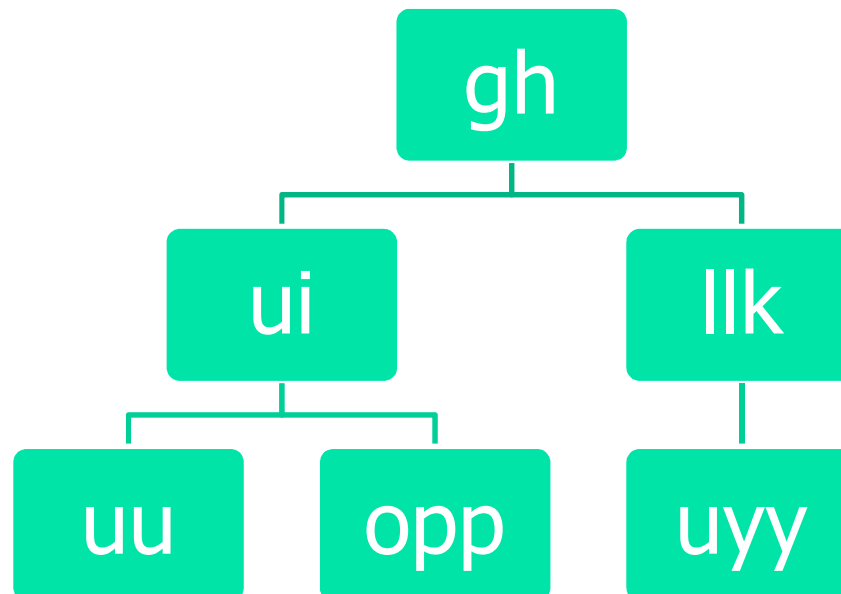
---

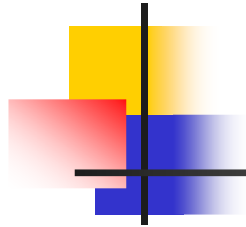
- Μαθηματικές μέθοδοι για τον υπολογισμό της απόδοσης αλγορίθμων και δομών δεδομένων.
- Θεωρητική Ανάλυση χωρίς υλοποίηση



## Βασικές Δομές Δεδομένων

- Θεμελιώδεις τρόποι οργάνωσης των δεδομένων
  - Σειριακή
  - Ιεραρχική
- Δομικά στοιχεία οργάνωσης προχωρημένων δομών





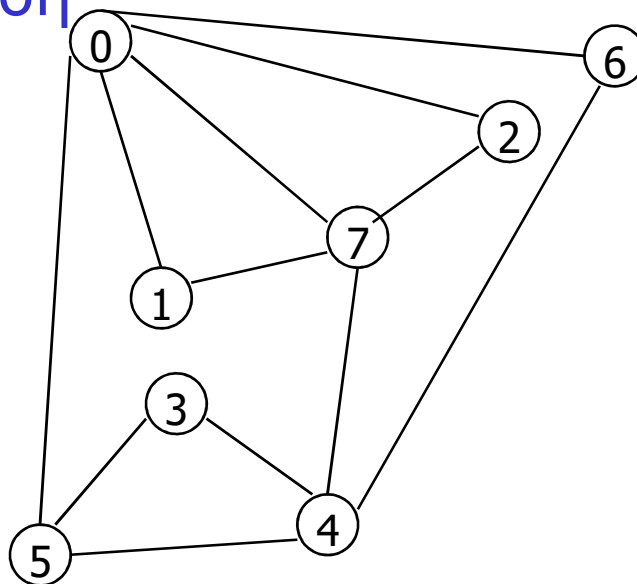
# Αναζήτηση και Ταξινόμηση

---

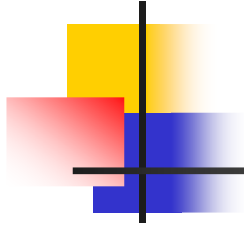
- Δομές Δεδομένων για γρήγορη αναζήτηση
  - Δένδρα Αναζήτησης
  - Πίνακες κατακερματισμού
- Αλγόριθμοι Ταξινόμησης
  - Στοιχειώδεις Αλγόριθμοι
  - Αποδοτικοί Αλγόριθμοι

# Γράφοι

- Δίκτυα κόμβων που συνδέονται μεταξύ τους
  - Συστηματική Εξερεύνηση
  - Εύρεση μονοπατιών
  - Ελάχιστη απόσταση



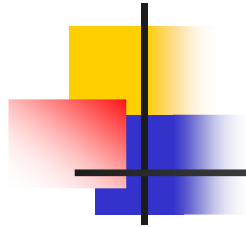




# Αφαίρεση

---

- Είναι γνωστό το τι μπορεί να κάνει ένας τύπος δεδομένων
- Το πώς το κάνει είναι κρυμμένο

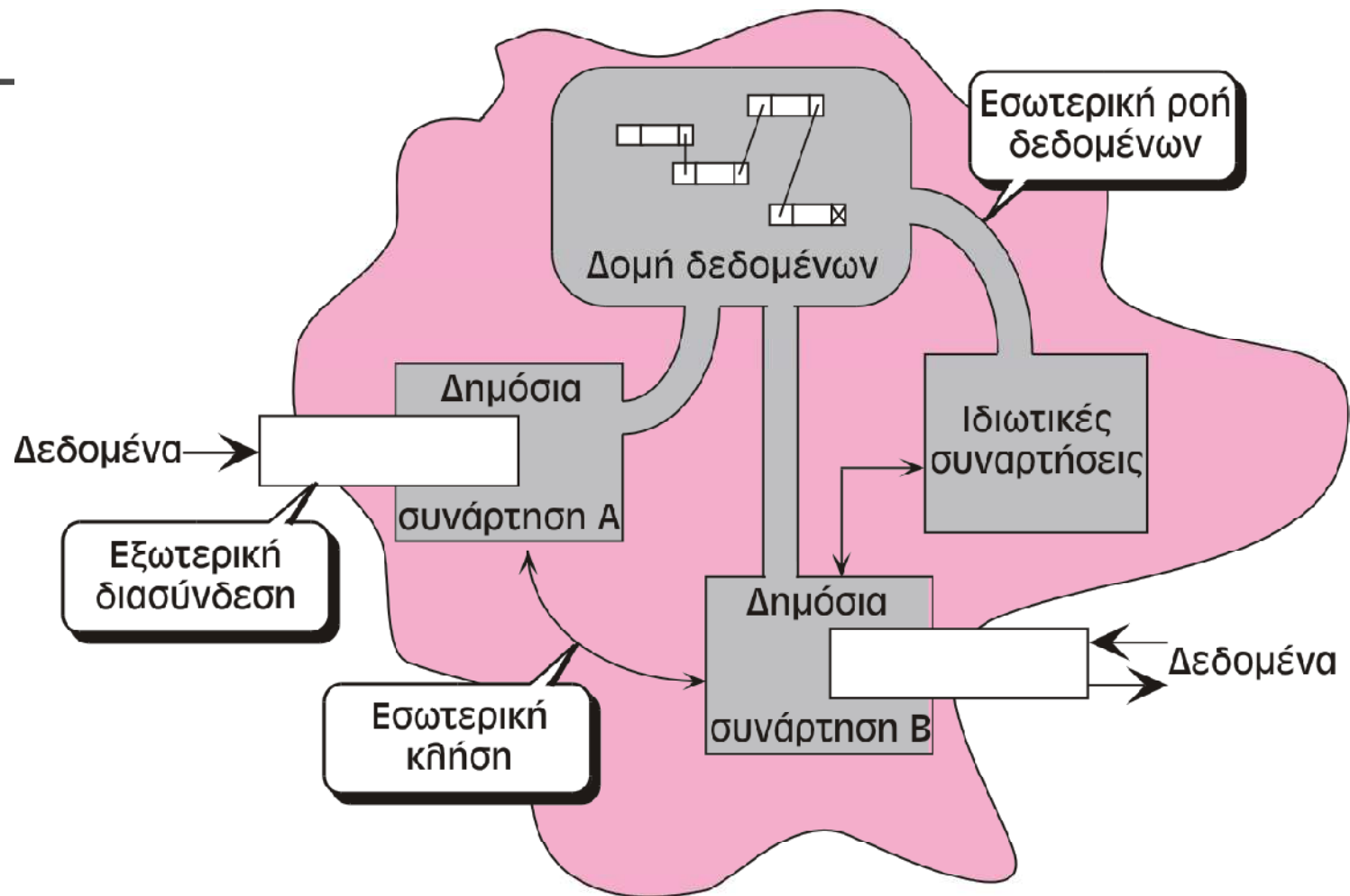
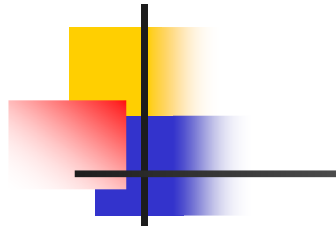


## Αφηρημένος Τύπος Δεδομένων


---

1. Δήλωση δεδομένων
2. Δήλωση λειτουργιών
3. Ενθυλάκωση δεδομένων και λειτουργιών

# Μοντέλο Αφηρημένου Τύπου Δεδομένων



# Καταχώρηση Δεδομένων

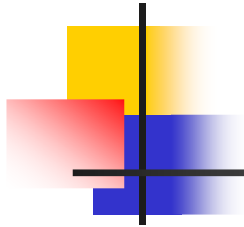
- 
- Τα δεδομένα αποθηκεύονται στη μνήμη
  - Θα κάνουμε τις εξής εξιδανικευμένες υποθέσεις
    - Έχουμε στη διάθεσή μας όση μνήμη χρειαζόμαστε
    - Το μέγεθος κάθε θέσης είναι ελαστικό ώστε να παριστάνει οποιαδήποτε πληροφορία μας ενδιαφέρει
    - Οι διευθύνσεις της μνήμης δεν είναι αριθμοί αλλά σύνθετα ονόματα όπως οι μεταβλητές σε μια γλώσσα προγραμματισμού



## Παράδειγμα, Αναζήτηση σε ένα κατάλογο ονομάτων

- Έστω ότι έχουμε μια λίστα από ονόματα (πχ. Πελάτες, υπάλληλοι κλπ. Πως όταν ξέρουμε το όνομα ή τον κωδικό ενός μπορούμε να εντοπίσουμε αν βρίσκεται στη λίστα και να εξετάσουμε τα υπόλοιπα στοιχεία του. Μια τέτοια λίστα είναι δυναμική και επομένως οι πράξεις που γίνονται είναι:
  - Εισαγωγή ονομάτων
  - Διαγραφή ονομάτων
  - Αναζήτηση

# Αναζήτηση

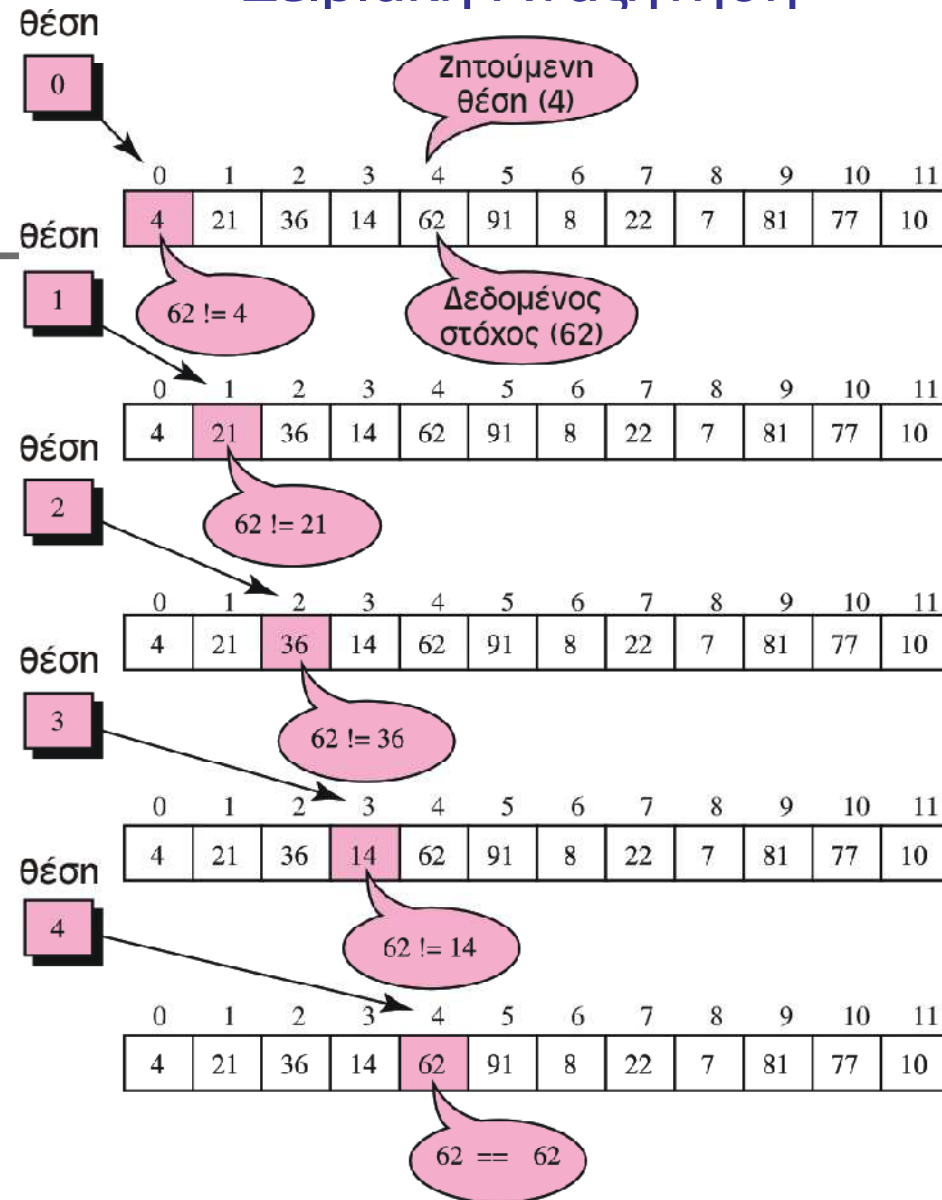
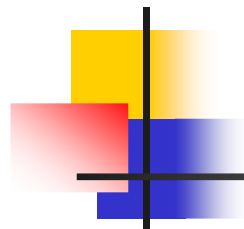


Ζητούμενη θέση

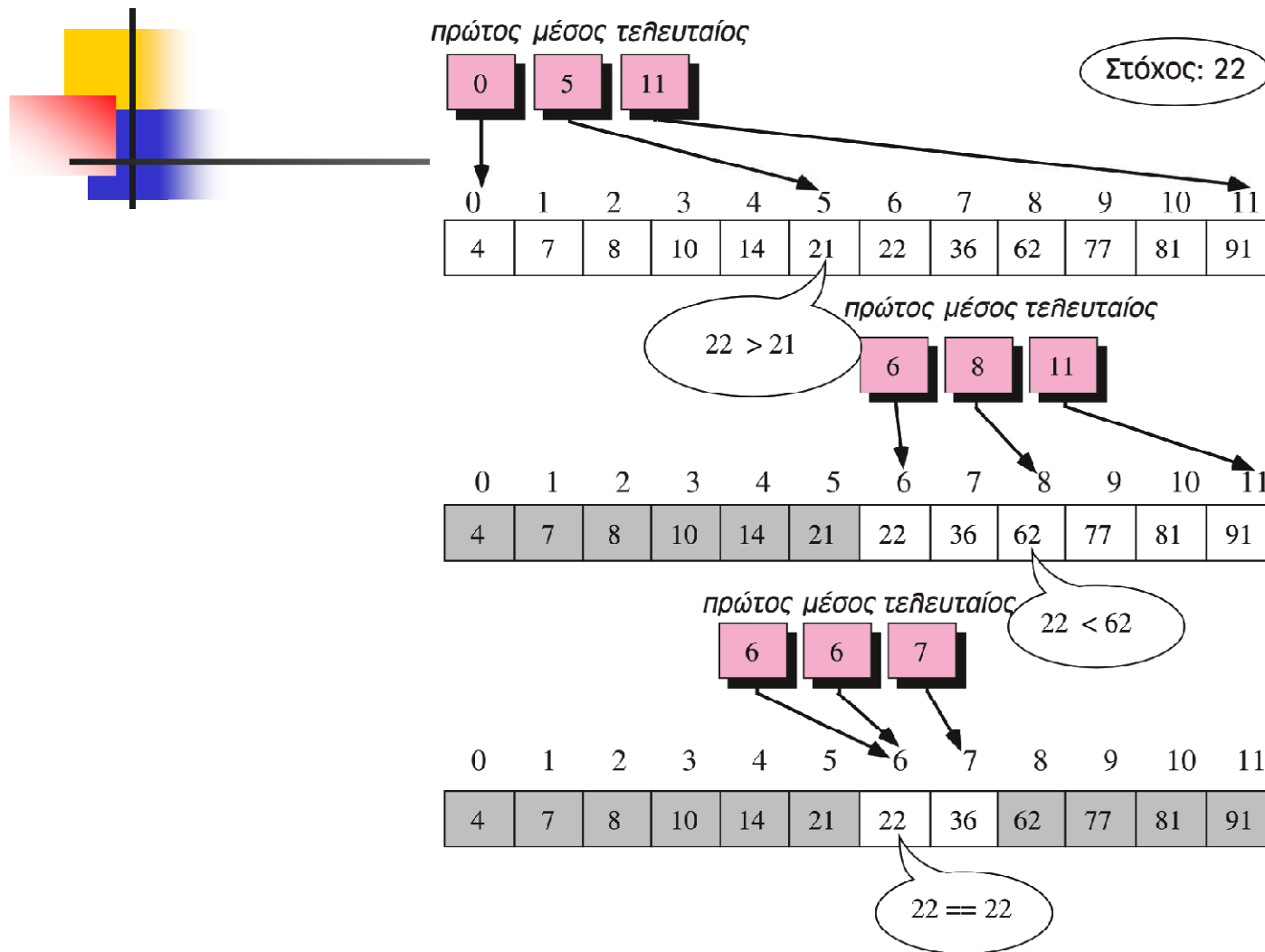
4	21	36	14	62	91	8	22	7	81	77	10
---	----	----	----	----	----	---	----	---	----	----	----

Δεδομένος  
στόχος (62)

# Σειριακή Αναζήτηση

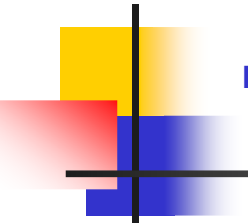


# Διαδική Αναζήτηση





# Παρατηρήσεις

- 
- Η ταχύτητα απάντησης οφείλεται στη δόμηση (ταξινομημένη) των δεδομένων και στη μεθοδολογία αναζήτησης που εφαρμόζεται ταξινομημένες λίστες. Για να μπορέσουμε να συνεχίσουμε να έχουμε τέτοια απόδοση θα πρέπει η συνθήκη (εδώ “ταξινόμηση”) να παραμένει αναλλοίωτη.
  - Σχεδίαση Δομής: σύλληψη ευέλικτου σχήματος καταχώρησης, διατήρηση κατάλληλης οργάνωσης των δεδομένων και συγγραφή αλγορίθμων που υλοποιούν τις λειτουργίες
  - Ανάλυση Δομής: απόδειξη πως διατηρείται η δομή, και υπολογισμός της επίδοσης

# Καταχώρηση Δεδομένων

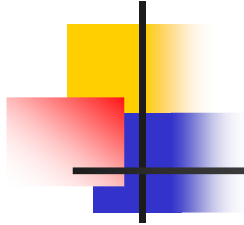


## ■ Θέσεις Μνήμης

- Κάθε δομένο καταχωρίζεται σε αντίστοιχη θέση μνήμης ή μεταβλητή
- Η διάκριση των δεδομένων και η πρόσβαση σε αυτά γίνεται με την αντιστοίχιση ενός ονόματος σε κάθε θέση μνήμης

## ■ Ενδιάμεσα Δεδομένα

- Κατά την διάρκεια υπολογισμών δημιουργούνται ενδιάμεσα αποτελέσματα, χωρίς να έχουμε πάντοτε την δυνατότητα ονομασίας τους



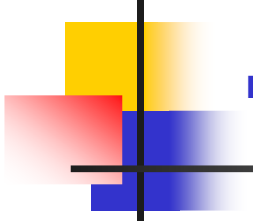
## Στατική Διαχείριση Μνήμης

- Χρησιμοποιείται όταν γνωρίζουμε το ανώτατο πλήθος θέσεων που θα χρειασθούμε από μια κατηγορία στοιχείων.
- Δίνουμε στα στοιχεία ένα όνομα και διακρίνουμε τα επιμέρους με ένα αύξοντα αριθμό

ΟΝΟΜΑ

1	Νίκος
2	Κώστας
3	Πέτρος
4	Θωμάς
5	Γιάννης

## Δυναμική Διαχείριση Μνήμη

- 
- Όταν δεν γνωρίζουμε το ανώτατο πλήθος θέσεων μνήμης που θα χρειασθούμε ή όταν αυτό μεταβάλλεται εντός ευρέων ορίων.
  - Η δυναμική διαχείριση της μνήμης γίνεται μέσω δεικτών. Οι δείκτες είναι δεδομένα με όνομα και διεύθυνση των οποίων η τιμή είναι όνομα-διεύθυνση μιας άλλης θέσης δεδομένων

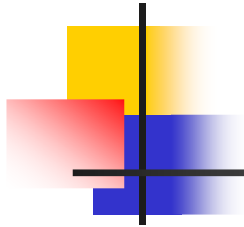
Συμβολισμός αν  $\Delta$  το όνομα ενός δείκτη το περιεχόμενο του είναι συμβολίζεται με  $\Delta^\wedge$  και είναι μια νέα θέση μνήμης.

## Διαδοχική καταχώρηση μνήμης

- Η υλοποίηση της στατικά ορισμένης μνήμης είναι σχεδόν ταυτόσημη με την διαδοχική καταχώρηση (πίνακας).
- Αριθμούνται  $N$  διαδοχικές θέσεις μνήμης  $T[1]$ ,  $T[2]$ , ...  $T[N]$

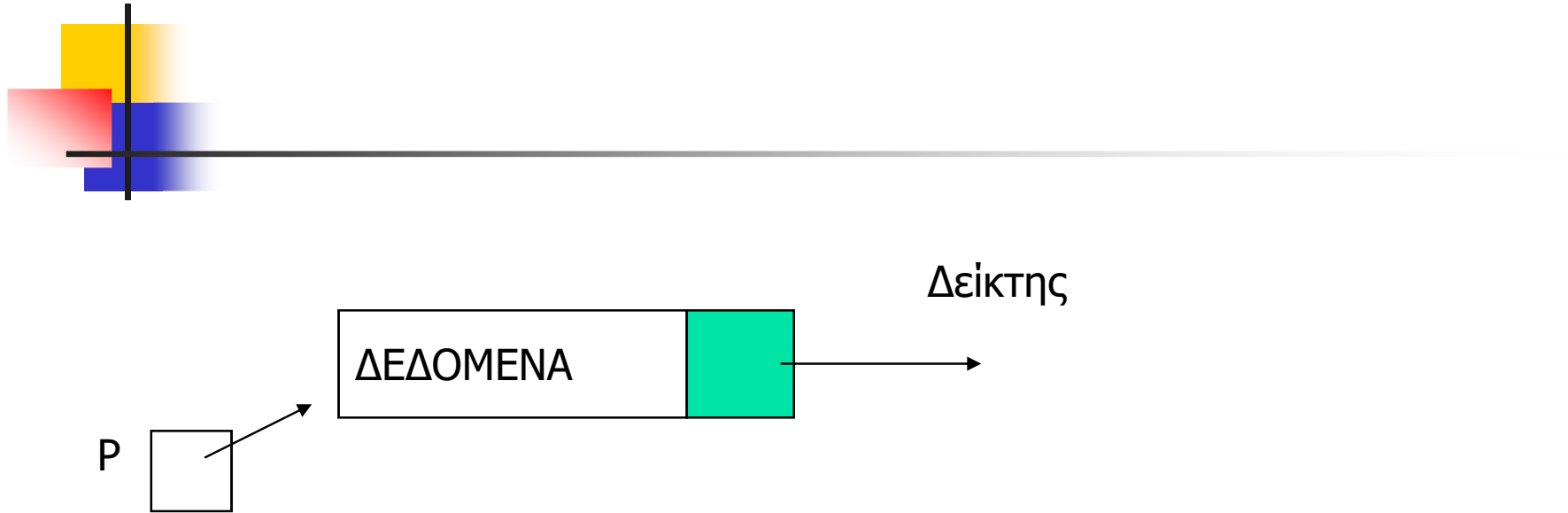
T	[1]	
	[2]	
	[3]	
	[4]	
	[5]	

# Αλυσιδωτή Καταχώρηση (1)

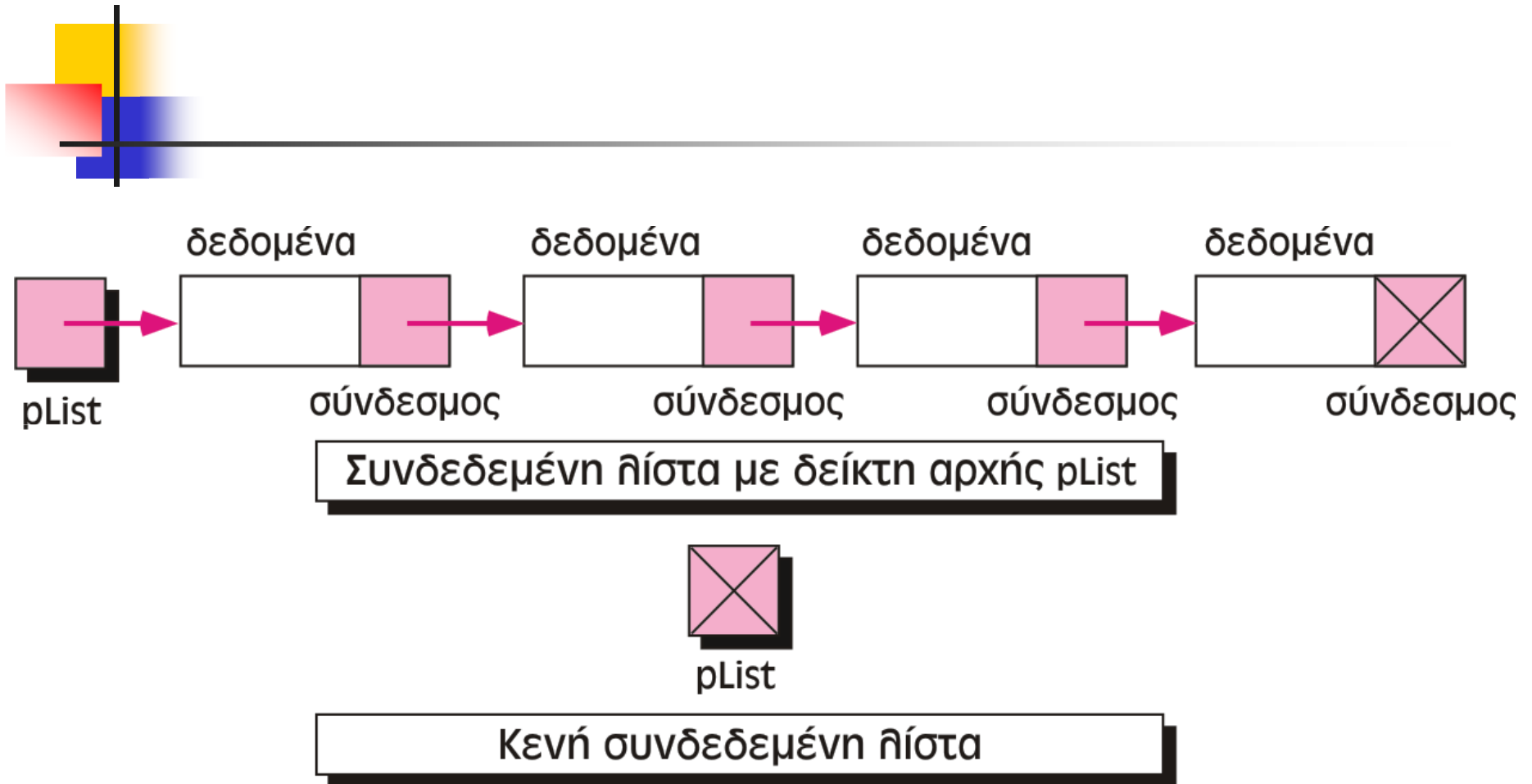


- Στη δυναμική διαχείριση μπορούμε να χρησιμοποιήσουμε μια θέση μνήμης όταν την χρειαζόμαστε υπό την προϋπόθεση ότι έχουμε ήδη ένα δείκτη σε αυτή. Δηλαδή θα πρέπει να προβλέψουμε ανάλογο πλήθος δεικτών.
- Μπορούμε στην ίδια θέση μνήμης που δημιουργείται να συμπεριλάβουμε ένα δείκτη σε μια περαιτέρω θέση μνήμης.
- Αυτό δημιουργεί μια αλυσίδα θέσεων μνήμης

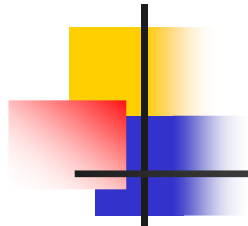
## Αλυσιδωτή Καταχώρηση (2)



# Αλυσιδωτή καταχώρηση





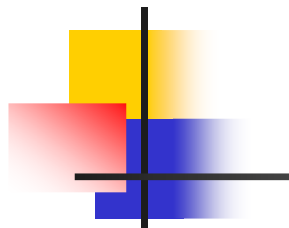


## Πίνακες

---

T↓

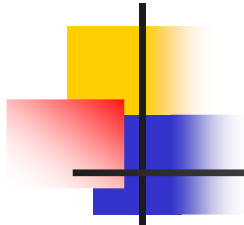
1	2	3	4	5	6	7	8	9	10
...	...	...	...	...	...	...	...	...	...



Πρώτη διάσταση  
(γραμμές)

0				
1				
2				
3				
4				
	0	1	2	3

Δεύτερη διάσταση  
(στήλες)



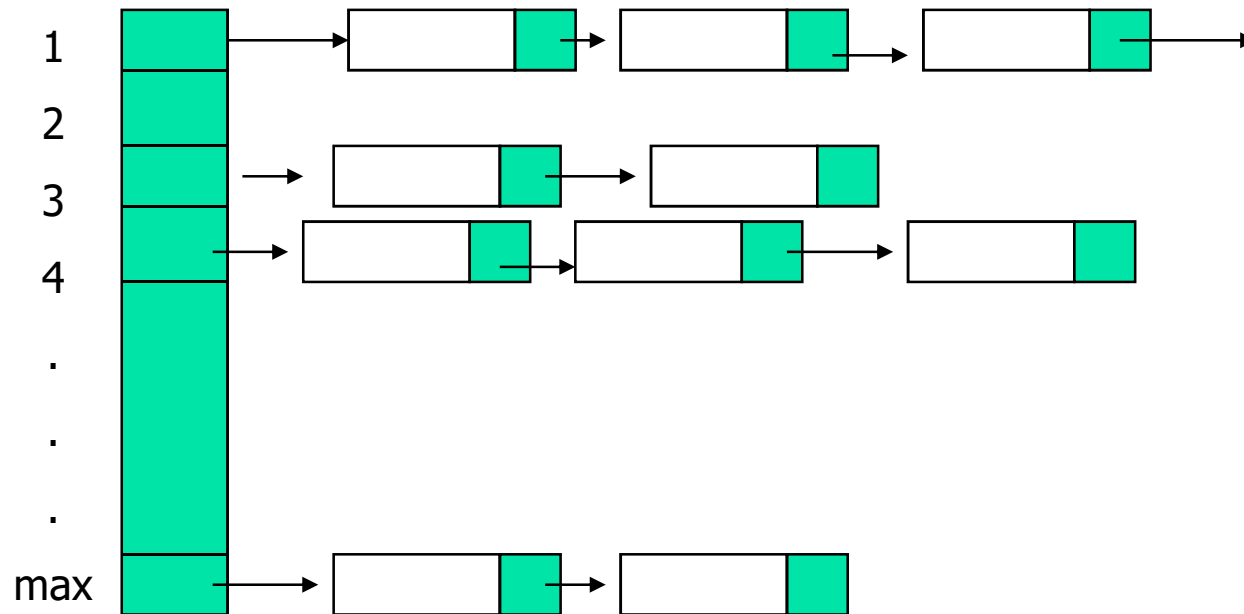
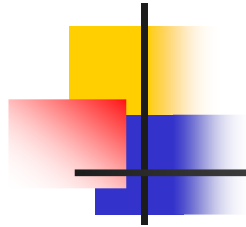
00	01	02	03	04
10	11	12	13	14

Άποψη χρήστη

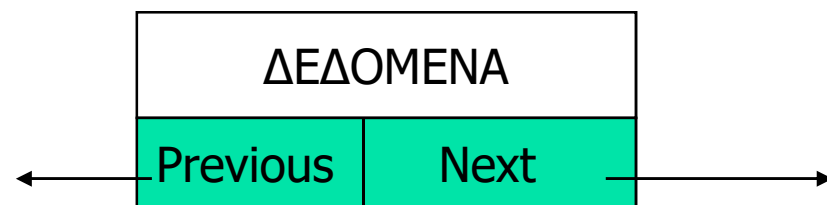
γραμμή 0					γραμμή 1				
00	01	02	03	04	10	11	12	13	14
[0][0]	[0][1]	[0][2]	[0][3]	[0][4]	[1][0]	[1][1]	[1][2]	[1][3]	[1][4]

Άποψη μνήμης

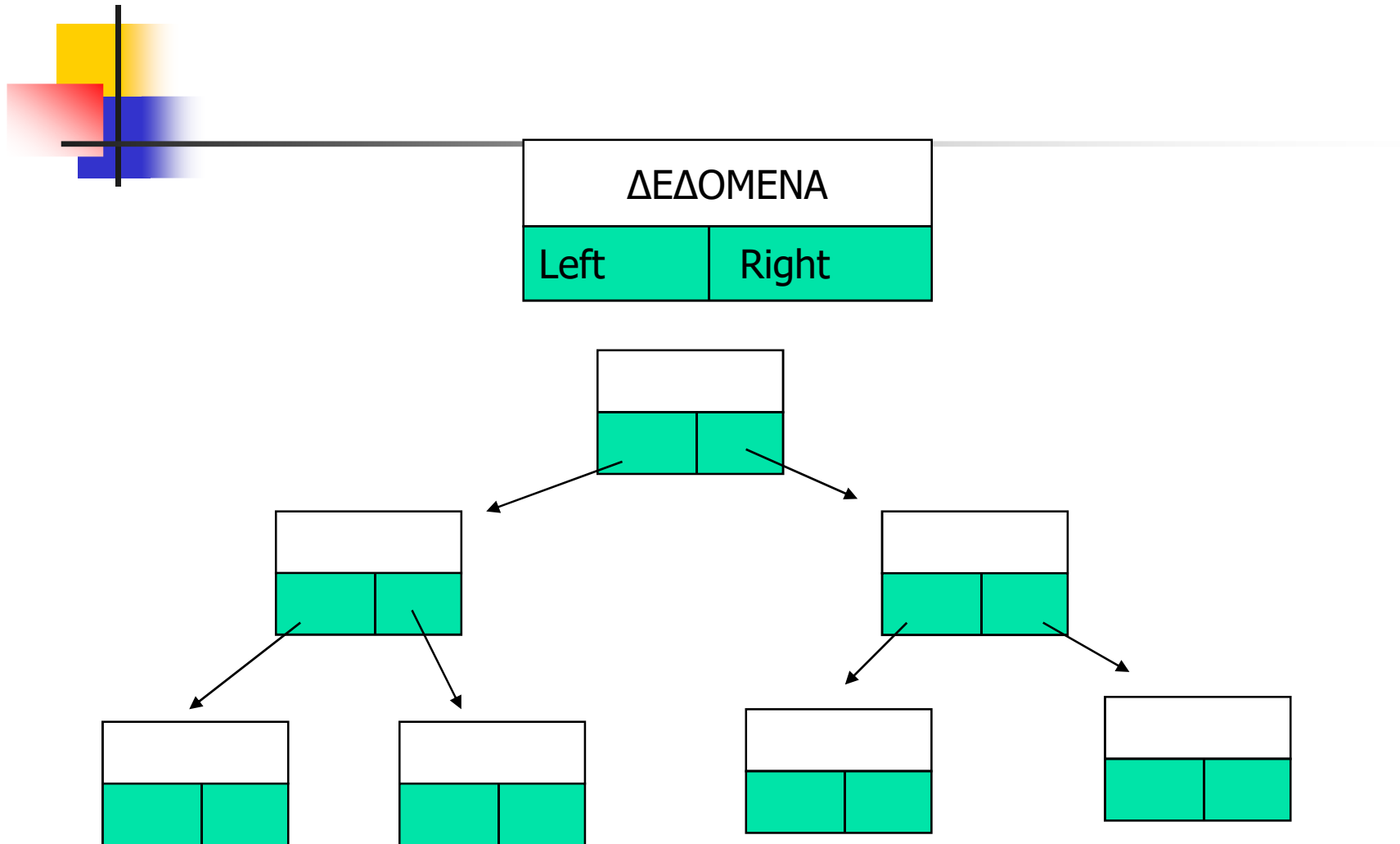
# Πίνακες από Αλυσίδες



# Διπλά Συνδεδεμένες Λίστες



# Δυαδικά Δένδρα



# Χειρισμοί Πινάκων (1)

Περιγραφή:

Table: όνομα πίνακα

Maxsize: μέγιστο πλήθος θέσεων

First: η πρώτη θέση με έγκυρη πληροφορία

Last: η τελευταία θέση με έγκυρη πληροφορία

Αρχική κατάσταση:

First:=0

Last:=0

Πλήθος στοιχείων ενός πίνακα:

Αν (Last=0 τότε Πλήθος:=0 διαφορετικά Πλήθος:= Last-First+1

## Χειρισμοί Πινάκων (2)

Εισαγωγή του στοιχείου  $\sigma$  στη θέση  $k$ :

Αν ( $\text{Last} < \text{Maxsize}$  και  $\text{First} \leq k$  και  $k \leq \text{Last}$ ) Τότε

{(Για  $i = \text{Last}$  έως  $k$

{  $\text{Table}[i+1] := \text{Table}[i]$ }

$\text{Table}[k] := \sigma$

$\text{Last} := \text{Last} + 1$ }