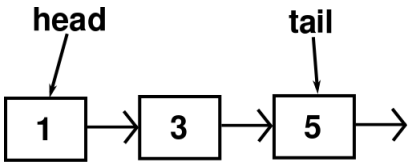
ARISTOTLE UNIVERSITY OF THESSALONIKI

SCHOOL OF INFORMATICS

DATA&WEB SCIENCE LABORATORY

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ



Συνδεδεμένες Λίστες

Απόστολος Ν. Παπαδόπουλος
Αναπληρωτής Καθηγητής
Τμήμα Πληροφορικής Α.Π.Θ.

Βασικές Έννοιες

Από τα προηγούμενα ξέρουμε ότι: τα στοιχεία ενός πίνακα (array) αποθηκεύονται συνεχόμενα στην κύρια μνήμη.

Η δομή της λίστα είναι πιο ευέλικτη. Τα στοιχεία της λίστας μπορούν να αποθηκεύονται **οπουδήποτε** στην κύρια μνήμη και στη γενική περίπτωση **δεν είναι συνεχόμενα** στη μνήμη.

Βασικές Έννοιες

Ομοειδή στοιχεία

- Αποτελούνται από 2 μέρη: **πληροφορία** και **δείκτη**.
  Η πληροφορία μπορεί να είναι ατομική ή ομαδική (εγγραφή).

Δυναμική δομή

- Μπορούμε να προσθαφαιρούμε στοιχεία χωρίς πρόβλημα.

Γραμμική δομή

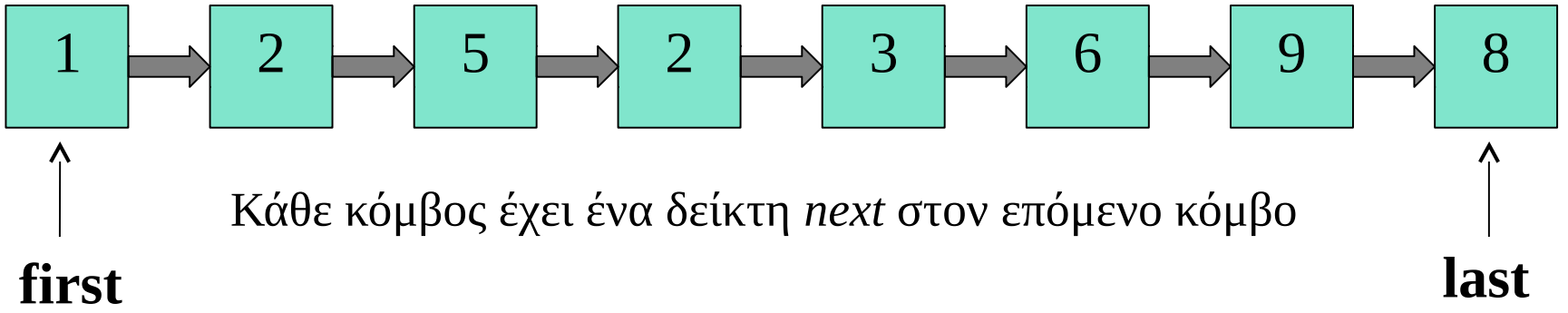
- Υπάρχει «λογική» διάταξη των στοιχείων, δηλαδή κάθε στοιχείο έχει ένα επόμενο (πλην του τελευταίου) και ένα προηγούμενο (πλην του πρώτου).

Αποθήκευση στοιχείων σε διάσπαρτες θέσεις

- Σειριακή προσπέλαση (ξεκινάμε πάντα από την αρχή και μετακινούμαστε μέσω των δεικτών στον επόμενο κόμβο).

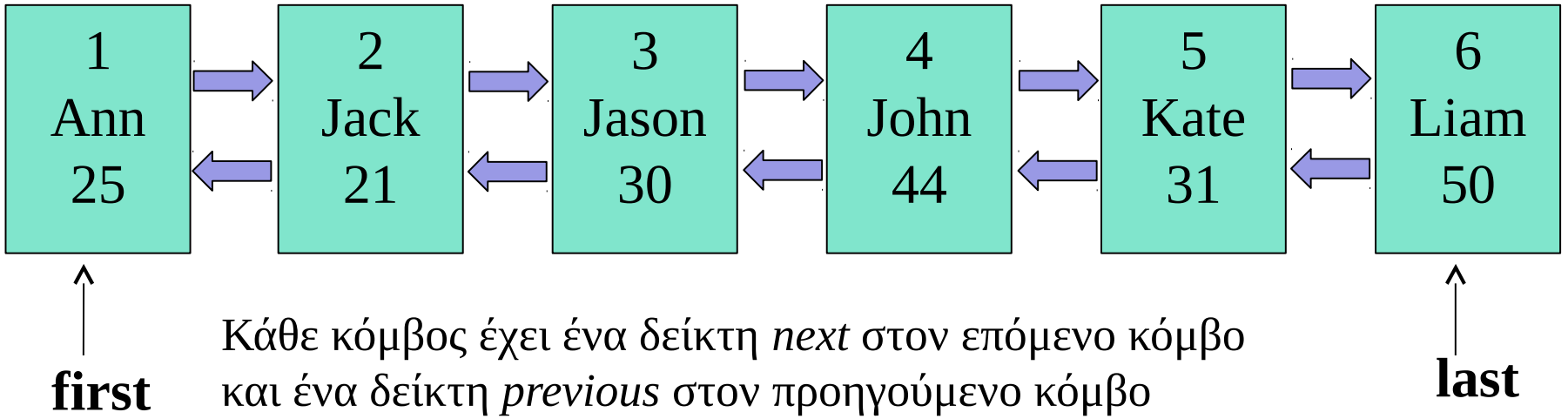
Βασικές Έννοιες

Συνδεδεμένη λίστα με ακεραίους



Κάθε κόμβος έχει ένα δείκτη *next* στον επόμενο κόμβο

Διπλά συνδεδεμένη λίστα με εγγραφές (σε αύξουσα σειρά ονομάτων)

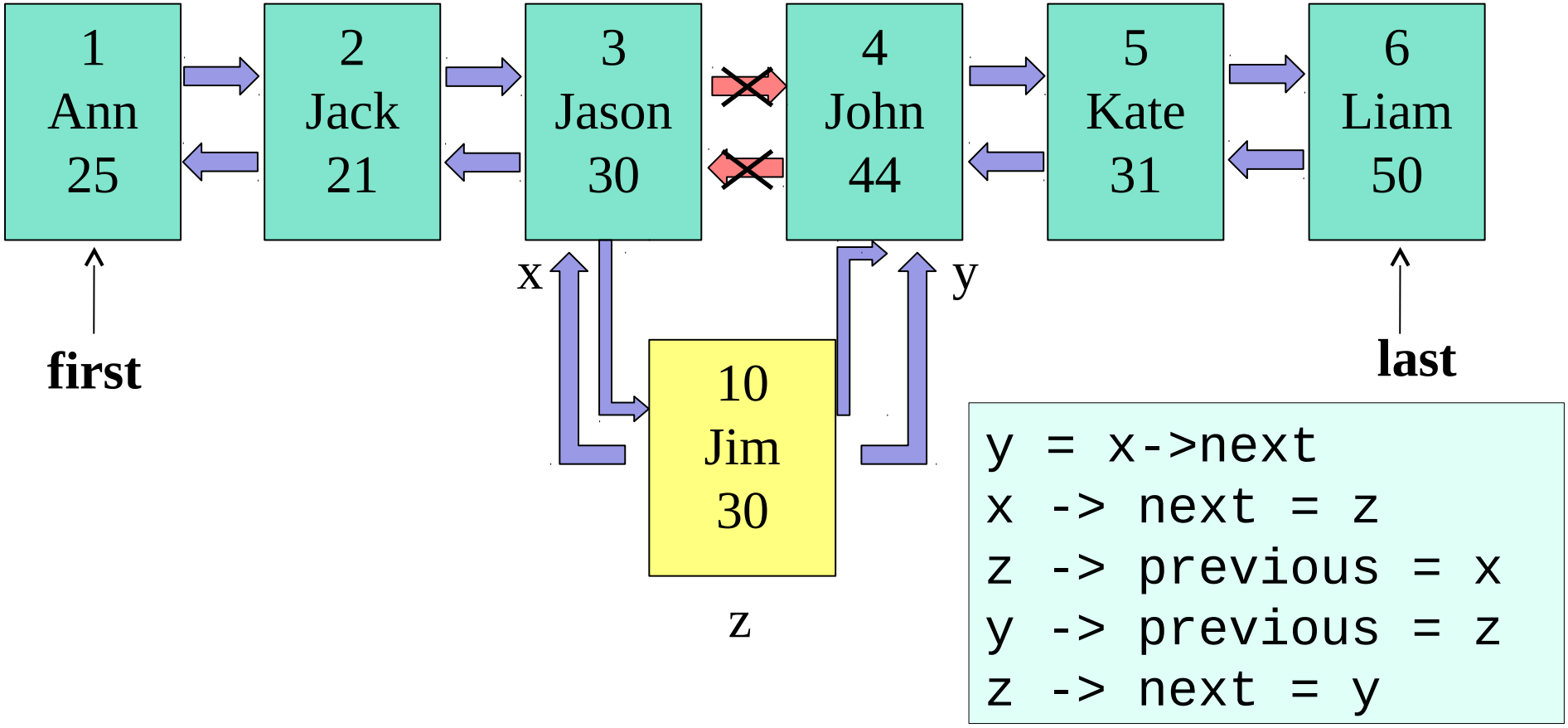


Κάθε κόμβος έχει ένα δείκτη *next* στον επόμενο κόμβο
και ένα δείκτη *previous* στον προηγούμενο κόμβο

Βασικές Έννοιες

Παράδειγμα εισαγωγής στοιχείου

(σε αύξουσα σειρά ονομάτων, αρχικά γνωστά x και z)



Βασικές Έννοιες

Η συνδεδεμένη λίστα είναι **δυναμική δομή**, με την έννοια ότι μπορούμε να μεγαλώνουμε τη λίστα χωρίς περιορισμούς.

Δεν απαιτείται να γνωρίζουμε το πλήθος των στοιχείων εξ αρχής (σε αντίθεση με την περίπτωση του πίνακα).

Βασικές Έννοιες

Για να μπορέσουμε να προσπελάσουμε ένα στοιχείο της λίστας, πρέπει να ξέρουμε το προηγούμενό του.

Η διάσχιση της λίστας ξεκινά πάντα από το πρώτο στοιχείο (first, head) και τερματίζει στο τελευταίο στοιχείο (last, tail).

Λειτουργίες

Αναζήτηση στοιχείου, O(*n*)

Εισαγωγή στοιχείου στην αρχή, O(1)

Εισαγωγή στοιχείου στο τέλος, O(1)

Εισαγωγή στοιχείου σε συγκεκριμένη θέση, O(*n*)

Διαγραφή head, O(1)

Διαγραφή tail, O(1)

Διαγραφή οποιουδήποτε στοιχείου, O(*n*)

Λειτουργίες

Μπορούμε να εκτελέσουμε δυαδική αναζήτηση σε συνδεδεμένη λίστα;

Λειτουργίες

Μπορούμε να εκτελέσουμε δυαδική αναζήτηση σε συνδεδεμένη λίστα;

Όχι, γιατί δεν μπορούμε να πάμε απευθείας στο μεσαίο στοιχείο (ισχύει και για τμήμα λίστας).

Άρα, η λίστα δεν είναι η καλύτερη δομή για αναζήτηση.

Χρήσεις

Η συνδεδεμένη λίστα αποτελεί πολύ καλή βάση για την υλοποίηση πιο πολύπλοκων δομών που απαιτούν **δυναμική δέσμευση μνήμης** όπως: στοίβα, ουρά, κλπ.

Ασκήσεις

Άσκηση 1

Δίνεται συνδεδεμένη λίστα και η θέση (pointer) σε κάποιο τυχαίο στοιχείο. Να διαγραφεί το στοιχείο στη θέση αυτή.

Προσοχή, **δεν έχουμε το δείκτη head**, παρά μόνο το δείκτη στο στοιχείο που θέλουμε να διαγράψουμε.

Παράδειγμα

Αν δίνεται η λίστα a → b → c → d → e και ο pointer p που δείχνει στο στοιχείο c, τότε το αποτέλεσμα πρέπει να είναι: a → b → d → e

Ασκήσεις

Άσκηση 2

Από μία συνδεδεμένη λίστα που είναι ταξινομημένη, να διαγραφούν οι πολλαπλές εμφανίσεις στοιχείων, διασχίζοντας τη λίστα μία φορά.

Παράδειγμα

Αν δίνεται η λίστα $1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 4 \rightarrow 4$ τότε το αποτέλεσμα πρέπει να είναι: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

Ασκήσεις

Άσκηση 3

Ταξινόμηση των στοιχείων της λίστας σε αύξουσα διάταξη.

Να περιγράψετε πως πρέπει να προσαρμοστεί η BubbleSort και η MergeSort ώστε να λειτουργούν με λίστα και όχι με πίνακα (array).

THANKS FOR LISTENING