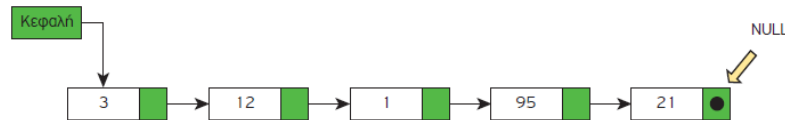


Τι ονομάζεται συνδεδεμένη λίστα;

Μία (απλά) **συνδεδεμένη λίστα** (linked list) είναι ένα σύνολο κόμβων διατεταγμένων γραμμικά (ο ένας μετά τον άλλο). Κάθε κόμβος περιέχει εκτός από τα **δεδομένα** του και έναν **δείκτη** που δείχνει προς τον **επόμενο** κόμβο.

Ο **δείκτης** του **τελευταίου** κόμβου δε δείχνει σε κάποιον κόμβο (δείκτης στο κενό). Για να το δηλώσουμε αυτό λέμε ότι το πεδίο δείκτη του τελευταίου κόμβου έχει την τιμή **NULL**.

**Πώς γίνεται η προσπέλαση σε μια συνδεδεμένη λίστα;**

Για να προσπελάσουμε τους κόμβους της λίστας χρειάζεται να γνωρίζουμε τη διεύθυνση (θέση στη μνήμη) του πρώτου κόμβου της λίστας. Η διεύθυνση αυτή αποθηκεύεται σε μία ειδική μεταβλητή που την ονομάζουμε συνήθως **Κεφαλή** (Head).

Οι κόμβοι μιας (απλά) συνδεδεμένης λίστας είναι διατεταγμένοι σε μια συγκεκριμένη σειρά, χωρίς αυτό να σημαίνει ότι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη. Αντίθετα, είναι **διασκορπισμένοι σε όλη τη μνήμη** και η σύνδεση μεταξύ τους γίνεται μέσω των δεικτών.

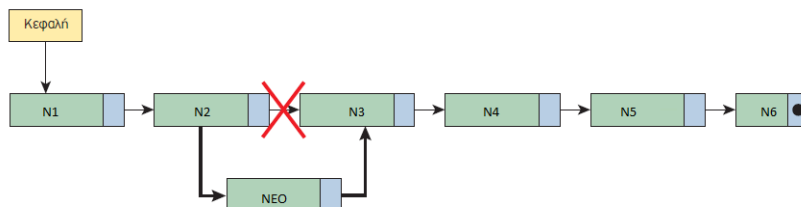
Έχουμε **άμεση πρόσβαση μόνο στον πρώτο κόμβο της λίστας**. Επομένως, για να εντοπίσουμε κάποιον από τους ενδιαμέσους κόμβους, πρέπει να ξεκινήσουμε από τον πρώτο κόμβο της λίστας και να ακολουθήσουμε τους δείκτες με τη σειρά, μέχρι να φτάσουμε στον επιθυμητό κόμβο (**Διάσχιση της Λίστας**)

Πώς προσθέτω κόμβο σε μια συνδεδεμένη λίστα;

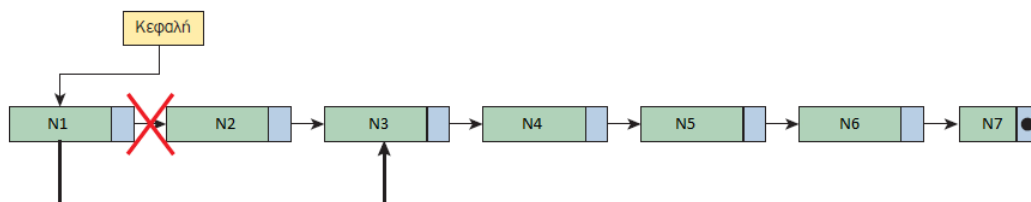
Οι απαιτούμενες ενέργειες για την εισαγωγή (παρεμβολή) του νέου κόμβου είναι :

- ο δείκτης του κόμβου μετά τον οποίο θα γίνει η παρεμβολή (N2), να δείχνει τον νέο κόμβο (NEO)
- και ο δείκτης του νέου κόμβου (NEO), να δείχνει αυτό που έδειχνε ο προηγούμενος (στο N3) (δηλαδή, να πάρει την τιμή που είχε πριν την εισαγωγή ο δείκτης του N2).

Με αυτόν τον τρόπο, οι κόμβοι της λίστας διατηρούν τη λογική τους σειρά, αλλά οι φυσικές θέσεις στη μνήμη μπορεί να είναι τελείως διαφορετικές.

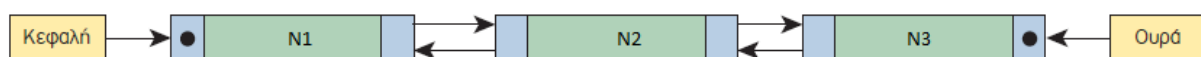
**Πώς διαγράφω κόμβο από μια συνδεδεμένη λίστα;**

Για τη διαγραφή ενός κόμβου αρκεί ν' αλλάξει τιμή ο δείκτης του προηγούμενου κόμβου και να δείχνει πλέον στον επόμενο αυτού που διαγράφεται. Ο κόμβος που διαγράφηκε (ο δεύτερος) αποτελεί «άχρηστο δεδομένο» και ο χώρος μνήμης που καταλάμβανε, παραχωρείται για άλλη χρήση.

**Τι είναι η διπλά συνδεδεμένη λίστα (doubly linked list);**

Η διπλά συνδεδεμένη λίστα είναι μια λίστα στην οποία μπορούμε να τη διατρέξουμε και προς τις δύο κατευθύνσεις. Αυτό επιτυγχάνεται με τη χρήση του δεύτερου δείκτη που δείχνει τον προηγούμενο κόμβο.

Προσφέρει έτσι τη δυνατότητα ξεκινώντας από οποιοδήποτε κόμβο της λίστας να μπορούμε να διατρέξουμε τη λίστα και προς τις δυο κατευθύνσεις.



Γιατί χρησιμοποιούνται οι **συνδεδεμένες λίστες** στην υλοποίηση της **στοίβας** και της **ουράς**

Οι συνδεδεμένες λίστες αξιοποιούνται για την υλοποίηση της στοίβας και της ουράς, λόγω της δυνατότητάς αυξομειώσης του μεγέθους τους.

Πώς μια **ΣΤΟΙΒΑ** μπορεί να υλοποιηθεί με μία **απλά συνδεδεμένη λίστα**;

Οι κόμβοι (στοιχεία) εισέρχονται από την αρχή της απλά συνδεδεμένης λίστας και εξέρχονται πάλι από την αρχή της λίστας τότε ο κόμβος (στοιχείο) που εισήλθε τελευταίος εξέρχεται και πρώτος (LIFO). Επομένως η στοίβα μπορεί να υλοποιηθεί με μία απλά συνδεδεμένη λίστα αφού μπορεί να γίνει εισαγωγή κόμβου (στοιχείου) στην αρχή της λίστας και διαγραφή κόμβου (στοιχείου) από την αρχή της λίστας .

Πότε θα χρησιμοποιήσουμε **ΛΙΣΤΑ** και πότε **ΠΙΝΑΚΑ** για την υλοποίηση μιας **ΣΤΟΙΒΑΣ**;

Καλύτερη είναι η υλοποίηση στοίβας με συνδεδεμένη λίστα εκτός αν δεν γνωρίζουμε εκ των προτέρων τον μέγιστο αριθμό στοιχείων και θέλουμε μια εύκολη και γρήγορη λύση, οπότε επιλέγουμε την υλοποίηση με πίνακα.

Πώς μια **ΟΥΡΑ** μπορεί να υλοποιηθεί με μία **διπλά συνδεδεμένη ΛΙΣΤΑ**.

Στην ουρά έχουμε τις εξής δύο κύριες λειτουργίες. Εισαγωγή στοιχείου στο πίσω άκρο της ουράς και εξαγωγή στοιχείου από το εμπρός άκρο της ουράς. Η ουρά μπορεί να υλοποιηθεί με μία διπλά συνδεδεμένη λίστα, αφού μπορούμε να κάνουμε εισαγωγή κόμβου στο τέλος της διπλά συνδεδεμένης λίστας. Επίσης μπορούμε να κάνουμε διαγραφή κόμβου στην αρχή της διπλά συνδεδεμένης λίστας (εξαγωγή στοιχείου από το εμπρός άκρο της ουράς).

Ποιες οι **διαφορές** ανάμεσα σε **ΛΙΣΤΕΣ** και **ΠΙΝΑΚΕΣ**;

- Ο **πίνακας** θεωρείται μια δομή **τυχαίας προσπέλασης**, σε αντίθεση με μια **λίστα** που είναι στην ουσία μια δομή ακολουθιακής ή **σειριακής** προσπέλασης. Για να φθάσουμε, δηλαδή, σ' έναν κόμβο μιας λίστας πρέπει να περάσουμε από όλους τους προηγούμενους ξεκινώντας από τον πρώτο.
- Ο **πίνακας** έχει **σταθερό** μέγεθος, το οποίο δηλώνεται εξαρχής κατά την υλοποίηση. Αυτό γίνεται, διότι ο πίνακας είναι στατική δομή δεδομένων σε αντίθεση με τη **λίστα** που είναι **δυναμική** δομή και το μέγεθός της μπορεί να μεταβάλλεται καθώς εισέρχονται νέοι κόμβοι στη λίστα ή διαγράφονται κάποιοι άλλοι.
- Οι κόμβοι της **λίστας** αποθηκεύονται σε **μη συνεχόμενες θέσεις** μνήμης σε αντιδιαστολή με τους **πίνακες**, όπου τα στοιχεία αποθηκεύονται σε **συνεχόμενες** θέσεις μνήμης.

Ποια είναι τα **πλεονεκτήματα των ΛΙΣΤΩΝ** (έναντι των ΠΙΝΑΚΩΝ) ;

- Το **δυναμικό** τους μέγεθος,
- η ευκολία **εισαγωγής** και **διαγραφής** από οποιοδήποτε μέρος της λίστας, καθώς και
- η **μη αναγκαιότητα δήλωσης του μεγέθους** τους.

Ποια τα **μειονεκτήματα των ΛΙΣΤΩΝ** (έναντι των ΠΙΝΑΚΩΝ);

- Η **τυχαία πρόσβαση** στη λίστα **δεν επιτρέπεται**. Είναι αδύνατο να φτάσετε στον n-οστό κόμβο μιας απλά συνδεδεμένης λίστας χωρίς πρώτα να περάσετε από όλους τους κόμβους διαδοχικά μέχρι τον συγκεκριμένο κόμβο ξεκινώντας από τον πρώτο κόμβο. Εναλλακτικά, στην περίπτωση της διπλά συνδεδεμένης λίστας μπορείτε να ξεκινήσετε και από τον τελευταίο κόμβο. Επομένως, **δεν μπορούμε** να πραγματοποιήσουμε με αποτελεσματικό τρόπο **δυναμική αναζήτηση** σε συνδεδεμένες λίστες.
- Οι συνδεδεμένες λίστες έχουν **πολύ μεγαλύτερη επιβάρυνση** από τους πίνακες, αφού οι συνδεδεμένοι κόμβοι της λίστας είναι δυναμικά κατανεμημένοι (οι οποίοι είναι λιγότερο αποτελεσματικοί στη **χρήση της μνήμης**) και κάθε κόμβος στη λίστα πρέπει, επιπλέον, να **αποθηκεύσει έναν πρόσθετο δείκτη** που θα δείχνει στον επόμενο κόμβο. Στην περίπτωση των διπλά συνδεδεμένων λιστών χρειαζόμαστε επιπλέον έναν δεύτερο δείκτη που θα δείχνει στον προηγούμενο κόμβο.

Ποιες είναι οι **βασικές ΠΡΑΞΕΙΣ** των **ΣΥΝΔΕΔΕΜΕΝΩΝ ΛΙΣΤΩΝ**;

- **Εισαγωγή** κόμβου στη λίστα (εισαγωγή κόμβου στην αρχή, στο τέλος της λίστας ή ενδιάμεσα).
- **Διαγραφή** κόμβου από τη λίστα (διαγραφή από την αρχή, το τέλος της λίστας ή ενδιάμεσα).
- **Έλεγχος** για το αν η λίστα είναι **κενή**.
- **Αναζήτηση** κόμβου για την εύρεση συγκεκριμένου στοιχείου.
- **Διάσχιση** της λίστας και προσπέλαση των στοιχείων της (π.χ. εκτύπωση των δεδομένων που περιέχονται σε όλους τους κόμβους της λίστας).

Τι ονομάζεται **ΔΕΝΔΡΟ**;

Ένα **δένδρο** (tree) είναι μία **δομή** που αποτελείται από ένα **σύνολο κόμβων** και ένα **σύνολο ακμών** μεταξύ των κόμβων με βάση τους εξής **κανόνες**:

- Υπάρχει ένας ξεχωριστός κόμβος που ονομάζεται **ρίζα**. Αυτός είναι ένας κόμβος **χωρίς γονέα**.
- Για κάθε κόμβο c , εκτός από τη ρίζα, υπάρχει **μόνο μια ακμή** που καταλήγει στον κόμβο αυτόν ξεκινώντας από κάποιον άλλον κόμβο p . Ο κόμβος p **ονομάζεται γονέας** του c και ο κόμβος c **παιδί** του p .
- Για κάθε κόμβο υπάρχει μία **μοναδική διαδρομή**, δηλαδή, μια ακολουθία διαδοχικών ακμών, που ξεκινάει από τη **ρίζα** και τερματίζει σε αυτόν τον **κόμβο**.

Δένδρο θεωρούμε και το κενό δένδρο, δηλαδή το δένδρο που δεν έχει ούτε κόμβους, ούτε ακμές. Το κενό δένδρο είναι το μόνο δένδρο χωρίς ρίζα.

Σε ένα **ΔΕΝΤΡΟ** ποιοι κόμβοι ονομάζονται **φύλλα** και ποιοι **αδέλφια**;

Οι κόμβοι χωρίς παιδιά ονομάζονται «**φύλλα**»

Κόμβοι με τον ίδιο γονέα ονομάζονται «**αδέλφια**».

Σε ποιους **τομείς της επιστήμης** χρησιμοποιούνται τα **ΔΕΝΤΡΑ**;

Τα δένδρα είναι μία μη-γραμμική ευέλικτη δομή δεδομένων που χρησιμοποιούνται σε πολλούς τομείς της επιστήμης των υπολογιστών, συμπεριλαμβανομένων των λειτουργικών συστημάτων, των γραφικών, των συστημάτων βάσεων δεδομένων, των παιχνιδιών, της τεχνητής νοημοσύνης και της δικτύωσης υπολογιστών.

Τα **δένδρα απόφασης**, είναι δένδρα στα οποία κάθε κόμβος αντιπροσωπεύει ένα χαρακτηριστικό (ιδιότητα), κάθε ακμή αντιπροσωπεύει μια απόφαση (κανόνα) και κάθε φύλλο αντιπροσωπεύει ένα αποτέλεσμα.

Στα παιχνίδια ο υπολογιστής χρησιμοποιεί ένα ειδικό δένδρο, που ονομάζεται **δένδρο του παιχνιδιού** (game tree), το οποίο μοντελοποιεί όλες τις πιθανές κινήσεις των παικτών.

Για **ποιους λόγους** τα **ΔΕΝΤΡΑ** θεωρούνται τα δένδρα **ισχυρές δομές**.

- Ο πρώτος λόγος αναφέρεται στη δυναμικότητα των δένδρων. Είναι πολύ εύκολο να προσθέσετε, να αφαιρέσετε ή να αναζητήσετε ένα στοιχείο σε ένα δένδρο.
- Ο δεύτερος βασικός λόγος είναι ότι η δομή των δένδρων μεταφέρει πληροφορίες.

Τι ονομάζεται **δυναμικό ΔΕΝΤΡΟ**;

Ένα **δυναμικό δένδρο** (binary tree) είναι ένα **διατεταγμένο** δένδρο, στο οποίο κάθε κόμβος έχει **το πολύ δύο** παιδιά, το αριστερό και το δεξί παιδί.

Μπορούμε, συνεπώς, να μιλάμε για αριστερό και δεξιό υποδένδρο ενός κόμβου.

Τι ονομάζεται **δυναμικό ΔΕΝΤΡΟ αναζήτησης**;

Ένα **δυναμικό δένδρο αναζήτησης** (binary search tree) είναι ένα **δυναμικό δένδρο**, όπου για **κάθε κόμβο** u , **όλοι οι κόμβοι του αριστερού υποδένδρου** έχουν τιμές **μικρότερες** της τιμής του κόμβου u και **όλοι οι κόμβοι του δεξιού υποδένδρου** έχουν τιμές **μεγαλύτερες** (ή ίσες) της τιμής του κόμβου u .

Τι ονομάζεται **ΓΡΑΦΟΣ** και ποιοι οι **τύποι του**;

Ένας **γράφος** (graph) είναι μία δομή που αποτελείται από ένα **σύνολο κόμβων** (ή σημείων ή κορυφών) και ένα σύνολο **γραμμών** (ή ακμών ή τόξων) που **ενώνουν** μερικούς ή όλους τους **κόμβους**. Ο γράφος αποτελεί την πιο γενική δομή δεδομένων, με την έννοια ότι **όλες οι προηγούμενες δομές** που παρουσιάστηκαν μπορούν να θεωρηθούν περιπτώσεις **γράφων**.

Εάν όλες οι **ακμές** σε έναν γράφο έχουν **κατεύθυνση**, ο γράφος ονομάζεται **κατευθυνόμενος γράφος** (directed graph).

Εάν όλες οι ακμές σε έναν γράφο δεν έχουν κατεύθυνση, ο γράφος ονομάζεται **μη κατευθυνόμενος γράφος** (undirected graph).