

## Στοιβά

Μία στοιβά δεδομένων μοιάζει με μία στοιβά από πιάτα. Για παράδειγμα, κάθε πιάτο που πλένεται τοποθετείται στη κορυφή (**top**) της στοιβάς των πιάτων, ενώ για σκούπισμα λαμβάνεται και πάλι το πιάτο της κορυφής. Αντίστοιχα, τα δεδομένα που βρίσκονται στην κορυφή της στοιβάς λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοιβάς λαμβάνονται τελευταία. Αυτή η μέθοδος επεξεργασίας ονομάζεται **Τελευταίο μέσα, Πρώτο έξω (LIFO- Last In First Out)**

Δύο είναι οι κύριες λειτουργίες σε μία στοιβά:

- Η **ώθηση (push)** στοιχείου στη κορυφή της στοιβάς, και
- Η **απώθηση (pop)** στοιχείου από τη στοιβά

Η διαδικασία της ώθησης πρέπει οπωσδήποτε να ελέγχει, αν η στοιβά είναι γεμάτη, οπότε τότε συμβαίνει **υπερχείλιση (overflow)** της στοιβάς. Αντίστοιχα, η διαδικασία απώθησης ελέγχει, αν υπάρχει ένα τουλάχιστον στοιχείο στη στοιβά, δηλαδή ελέγχει αν γίνεται **υποχείλιση (underflow)** της στοιβάς.

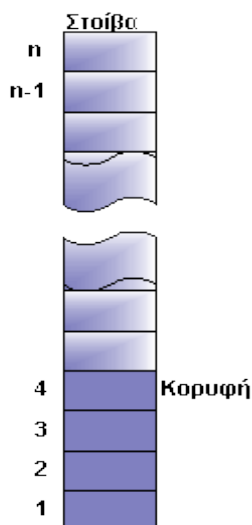
Μία στοιβά μπορεί να υλοποιηθεί πολύ εύκολα με τη βοήθεια ενός μονοδιάστατου πίνακα. Μια βοηθητική μεταβλητή (με το όνομα συνήθως **top**) χρησιμοποιείται για να δείχνει το στοιχείο που τοποθετήθηκε τελευταίο στην κορυφή της στοιβάς.

Για την **ΩΘΗΣΗ** ενός νέου στοιχείου στη στοιβά

- Ελέγχουμε για υπερχείλιση **top < n**
- Αυξάνεται η μεταβλητή **top** κατά ένα (**top ← top + 1**)
- Στη θέση αυτή να εισέρχεται το στοιχείο.

Για την **ΑΠΩΘΗΣΗ** ενός στοιχείου από τη στοιβά

- Ελέγχουμε για υποχείλιση **top > 0**
- Εξέρχεται το στοιχείο που δείχνει η μεταβλητή **top**
- Και στη συνέχεια μειώνεται η μεταβλητή **top** κατά ένα (**top ← top - 1**) για να δείχνει τη νέα κορυφή.



## Ουρά

Οι ουρές είναι καθημερινό φαινόμενο. Για παράδειγμα, ουρές δημιουργούνται όταν άνθρωποι, αυτοκίνητα, εργασίες, προγράμματα κ.λπ. περιμένουν για να εξυπηρετηθούν. Σε μία ουρά αναμονής με ανθρώπους, συμβαίνει να εξυπηρετείται εκείνος που στάθηκε στην ουρά πρώτος απ' όλους τους άλλους. Η μέθοδος αυτής επεξεργασίας ονομάζεται **Πρώτο μέσα πρώτο έξω (FIFO-First In First Out)**.

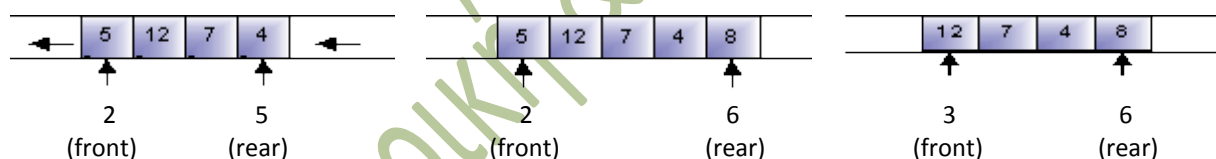
Δυο είναι οι κύριες λειτουργίες που εκτελούνται σε μία ουρά:

- Η **εισαγωγή (enqueue)** στοιχείου στο πίσω της ουράς, και
- Η **εξαγωγή (dequeue)** στοιχείου από το εμπρός άκρο της ουράς

Σε κάθε περίπτωση πρέπει να ελέγχεται πριν από κάθε ενέργεια αν υπάρχει ελεύθερος χώρος στον πίνακα για την εισαγωγή ή αν υπάρχει τουλάχιστον ένα στοιχείο για εξαγωγή

Άρα, σε αντίθεση με τη δομή της στοιβάς, στην περίπτωση της ουράς απαιτούνται δυο δείκτες: ο **εμπρός (front)** και ο **πίσω (rear)** δείκτης, που μας δίνουν τη θέση του στοιχείου που σε πρώτη ευκαιρία θα εξαχθεί και τη θέση του στοιχείου που μόλις εισήλθε.

Στο παρακάτω σχήμα φαίνεται μια ουρά με τέσσερα στοιχεία στην οποία εισάγεται ένα στοιχείο και ακολούθως εξαγεται ένα στοιχείο



α) Μια ουρά με 4 στοιχεία

β) Η ουρά μετά εισαγωγή του στοιχείου 8

γ) Η ουρά μετά την εξαγωγή του στοιχείου 5

Μια ουρά μπορεί να υλοποιηθεί με τη βοήθεια ενός μονοδιάστατου πίνακα.

Για την ΕΙΣΑΓΩΓΗ ενός στοιχείου στην ουρά (Πίσω)

- Ελέγχουμε αν υπάρχει θέση στην ουρά (**rear < n**)
- Αυξάνεται ο δείκτης rear κατά ένα (**rear ← rear + 1**)
- Στη θέση αυτή αποθηκεύεται το στοιχείο.

Για την ΕΞΑΓΩΓΗ, ενός στοιχείου από την ουρά (Εμπρός)

- Ελέγχουμε αν υπάρχει ένα τουλάχιστον στοιχείο για να πάρουμε (**front ≤ rear**)
- Εξέρχεται το στοιχείο που δείχνει ο δείκτης front.
- Αυξάνεται ο δείκτης front κατά ένα (**front ← front + 1**), για να δείχνει το επόμενο στοιχείο που πρόκειται να εξαχθεί.