
Γ Λυκείου

Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον

ΣΤΟΙΒΑ - ΟΥΡΑ

Περιεχόμενα

Κεφάλαιο 3 (Βιβλίο Ι): Δομές Δεδομένων και αλγόριθμοι

- 3.4 Στοίβα
- 3.5 Ουρά

Κεφάλαιο 1 (Βιβλίο ΙΙ): Δομές Δεδομένων και Αλγόριθμοι

- 1.1 Στοίβα
- 9.2 Ουρά

Κατηφόρης Παναγιώτης
2019-2020

Περιγραφή της στοιίβας.

Μία στοιίβα δεδομένων μοιάζει με μία στοιίβα από πιάτα. Για παράδειγμα κάθε πιάτο που πλένεται τοποθετείται στη κορυφή (top) της στοιίβας των πιάτων, ενώ για σκούπισμα λαμβάνεται και πάλι το πιάτο της κορυφής.

Σε μια στοιίβα το ένα αντικείμενο είναι πάνω στο άλλο. Έτσι αν θέλουμε να προσθέσουμε (**ώθηση**) ένα νέο αντικείμενο αυτό θα τοποθετηθεί πάντα στην κορυφή της στοιίβας. Αντίστοιχα, αν θέλουμε να αφαιρέσουμε (**απώθηση**) ένα αντικείμενο από μια στοιίβα, θα πρέπει να αφαιρέσουμε ένα-ένα όλα τα αντικείμενα που το καλύπτουν. Μάλιστα η απώθηση γίνεται από την κορυφή της στοιίβας. Έτσι τα δεδομένα που βρίσκονται στην κορυφή της στοιίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοιίβας λαμβάνονται τελευταία.

Μέθοδος επεξεργασίας της στοιίβας

Αυτή η μέθοδος επεξεργασίας ονομάζεται **LIFO** από τα αρχικά των λέξεων **Last In First Out** που σημαίνει **Τελευταίο μέσα - Πρώτο έξω**.

Ορισμός

Στοιίβα (stack), ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που βρίσκονται στην κορυφή της στοιίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοιίβας λαμβάνονται τελευταία.

Λειτουργίες της στοιίβας - Υπερχείλιση & Υποχείλιση

Δύο είναι οι κύριες λειτουργίες σε μία στοιίβα:

- ✚ Η **ώθηση (push)** στοιχείου **στη κορυφή** (top) της στοιίβας
Κατά την διαδικασία της ώθησης πρέπει οπωσδήποτε να ελέγχετε, αν η στοιίβα είναι γεμάτη. Στην περίπτωση που προσπαθήσουμε να «ωθήσουμε» ένα νέο στοιχείο σε μία γεμάτη στοιίβα τότε συμβαίνει **υπερχείλιση** (overflow) της στοιίβας
- ✚ Η **απώθηση (pop)** στοιχείου **από τη κορυφή** (top) στοιίβα
Αντίστοιχα, κατά την διαδικασία της **απώθησης** ελέγχετε, αν υπάρχει ένα τουλάχιστον στοιχείο στη στοιίβα. Στην περίπτωση που προσπαθήσουμε να «αφαιρέσουμε» ένα στοιχείο από μία κενή στοιίβα, έχουμε **υποχείλιση** (underflow) της στοιίβας

Παρατηρήσεις

Στη δομή δεδομένων **στοίβα** παρατηρούμε τα εξής:

1. Η μέθοδος επεξεργασίας της στοιίβας είναι η **LIFO** (Last In First Out)
2. Δύο είναι η βασικές επεξεργασίες που μπορούμε να κάνουμε σε μια στοιίβα:
 - i. Η **ώθηση** ενός νέου στοιχείου στην κορυφή της στοιίβας.
 - ii. Η **απώθηση** του στοιχείου που βρίσκεται στην κορυφή της στοιίβας
3. Κάθε νέο στοιχείο προστίθεται στην **κορυφή** της στοιίβας.
4. Κάθε φορά μπορούμε να επεξεργαστούμε **μόνο** το στοιχείο που και συγκεκριμένα το στοιχείο που βρίσκεται στην κορυφή της.
5. Για να επεξεργαστούμε ένα στοιχείο που **δεν** βρίσκεται στην κορυφή της στοιίβας πρέπει να απωθήσουμε όλα τα στοιχεία που βρίσκονται από πάνω του.
6. Μπορεί να γίνει **απώθηση**, από την **κορυφή** της στοιίβας, μόνο **ενός** στοιχείου κάθε φορά.

Υλοποίηση της στοίβας με τη βοήθεια πίνακα

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

Για την εισαγωγή ενός νέου στοιχείου στη στοίβα (**ώθηση**) αρκεί να αυξηθεί η μεταβλητή top κατά ένα και στη θέση αυτή να εισέλθει το στοιχείο (αφού ελέγξουμε μήπως η στοίβα είναι γεμάτη, δηλαδή μήπως έχουμε **υπερχείλιση**).

Αντίθετα για την εξαγωγή ενός στοιχείου από τη στοίβα (**απόθεση**) (αφού ελέγξουμε ότι στην στοίβα υπάρχει τουλάχιστον ένα στοιχείο δηλαδή δεν έχουμε **υποχείλιση**) «απωθείται» πρώτα το στοιχείο που δείχνει η μεταβλητή top και στη συνέχεια η top μειώνεται κατά ένα για να δείχνει τη νέα κορυφή.

Παράδειγμα

N		
N-1		
...		
3		
2	35	←top
1	12	

Χρησιμοποιούμε πίνακα N θέσεων. Η μεταβλητή **top** δείχνει την θέση του στοιχείου που **τοποθετήθηκε τελευταίο** στην κορυφή της στοίβας.
Δηλ. top=2

N		
N-1		
...		
3	56	←top
2	35	
1	12	

Για την **ώθηση** ενός νέου στοιχείου στη στοίβα (56) **αυξάνουμε** το δείκτη top κατά 1 (top = 3) και τοποθετούμε το στοιχείο (56) στη θέση που δείχνει ο top (δηλ. στη θέση 3). Πριν την ώθηση ελέγχουμε αν γίνεται **υπερχείλιση**.

N		
N-1		
...		
3	56	
2	35	←top
1	12	

Για την **απόθεση** στοιχείου από τη στοίβα, εξέρχεται το στοιχείο που δείχνει ο top (δηλ το 56) και στην συνέχεια ο top **μειώνεται** κατά 1 (δηλ. top =2). Πριν την απόθεση ελέγχουμε αν γίνεται **υποχείλιση**.

Ωθηση - Απόθεση

Στοιβα[] : το όνομα του πίνακα με τον οποίο υλοποιούμε την στοίβα

N : το μέγεθος του πίνακα

top : η μεταβλητή που δείχνει το πάνω στοιχείο της στοίβας

ΩΘΗΣΗ

```
ΓΡΑΨΕ 'Δώσε τιμή για εισαγωγή'
ΔΙΑΒΑΣΕ νέο_στοιχείο
! αρχικά ελέγχεται αν η στοίβα είναι γεμάτη
(υπερχείλιση)
ΑΝ top < N ΤΟΤΕ
! αν δεν είναι αυξάνουμε τον top κατά 1
top←top+1
! εισάγουμε το νέο στοιχείο στην στοίβα
στοιβα[top] ← νέο_στοιχείο
ΑΛΛΙΩΣ
! αν η στοίβα είναι γεμάτη...
ΓΡΑΨΕ 'αδύνατη η ώθηση λόγω'
ΓΡΑΨΕ 'υπερχείλισης'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΑΠΩΘΗΣΗ

```
! αρχικά ελέγχεται αν η στοίβα περιέχει έστω
και ένα στοιχείο (υποχείλιση)
ΑΝ top > 0 ΤΟΤΕ
! αν ναι εξάγεται το πάνω-πάνω στοιχείο
ΓΡΑΨΕ στοιβα[top]
! και μειώνεται ο top κατά 1
top←top-1
ΑΛΛΙΩΣ
! αν η στοίβα είναι άδεια...
ΓΡΑΨΕ 'αδύνατη η απόθεση λόγω'
ΓΡΑΨΕ 'υπορχείλισης'
ΤΕΛΟΣ_ΑΝ
```

Περιγραφή της ουράς

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.

Σε μια ουρά το ένα αντικείμενο είναι πίσω από το άλλο. Έτσι αν θέλουμε να προσθέσουμε (**εισαγωγή**) ένα νέο αντικείμενο αυτό θα τοποθετηθεί πάντα πίσω από το τελευταίο αντικείμενο που έχει ήδη εισαχθεί στην ουρά. Αντίστοιχα, αν θέλουμε να αφαιρέσουμε (**εξαγωγή**) ένα αντικείμενο από μια ουρά, θα πρέπει να αφαιρέσουμε ένα-ένα όλα τα αντικείμενα που βρίσκονται μπροστά από αυτό.

Μέθοδος επεξεργασίας της ουράς

Αυτή η μέθοδος επεξεργασίας ονομάζεται **FIFO** από τα αρχικά των λέξεων **First In First Out** που σημαίνει **Πρώτο μέσα - Πρώτο έξω**.

Ορισμός

Ουρά (Queue), ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα.

Λειτουργίες της ουράς

Δυο είναι οι κύριες λειτουργίες που εκτελούνται σε μία ουρά:

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

Σε κάθε περίπτωση όμως, πρέπει να ελέγχεται πριν από οποιαδήποτε ενέργεια, αν υπάρχει ελεύθερος χώρος στον πίνακα για την εισαγωγή και αν υπάρχει ένα τουλάχιστον στοιχείο για την εξαγωγή.

Οι δύο δείκτες της ουράς

Σε αντίθεση με τη δομή της στοίβας, στην περίπτωση της ουράς απαιτούνται δυο δείκτες για την επεξεργασία της:

- ✚ ο δείκτης **εμπρός (front)** που μας δίνει τη θέση του στοιχείου που σε πρώτη ευκαιρία θα εξαχθεί και
- ✚ ο δείκτης **πίσω (rear)** που μας δίνει τη θέση του στοιχείου που μόλις εισήλθε.

Παράδειγμα

← 5 | 12 | 7 | 4 | ←

↑ ↑
εμπρός **πίσω**
(front) **(rear)**

Το στοιχείο 5 είναι σε «θέση εξόδου». Το στοιχείο 4 «μόλις μπήκε».

5 | 12 | 7 | 4 | 8

↑ ↑
εμπρός **πίσω**
(front) **(rear)**

Η ουρά μετά την εισαγωγή του στοιχείου 8.

5 | 12 | 7 | 4 | 8

↑ ↑
εμπρός **πίσω**
(front) **(rear)**

Η ουρά μετά την εξαγωγή του 5.

Εισαγωγή - Εξαγωγή

ουρά[] : το όνομα του πίνακα με τον οποίο υλοποιούμε την ουρά.

N : το μέγεθος του πίνακα

front : η μεταβλητή που δείχνει το εμπροσθεν στοιχείο της ουράς (προς εξαγωγή)

rear : η μεταβλητή που δείχνει το τελευταίο στοιχείο της ουράς (μόλις εισήχθη)

ΕΙΣΑΓΩΓΗ

```
ΓΡΑΨΕ 'Δώσε τιμή για εισαγωγή'
ΔΙΑΒΑΣΕ νέο_στοιχείο
! αρχικά ελέγχεται αν η ουρά είναι γεμάτη
ΑΝ rear = N ΤΟΤΕ
    ! αν ναι τότε εμφανίζεται σχετικό μήνυμα
    ΓΡΑΨΕ 'η ουρά είναι γεμάτη'
    ! ελέγχεται αν η ουρά είναι άδεια
    ΑΛΛΙΩΣ_ΑΝ (front=0 ΚΑΙ rear=0) ΤΟΤΕ
        ! οπότε αρχικοποιούμε τους 2 δείκτες
        front ← 1
        rear ← 1
        ! και εισάγουμε το στοιχείο
        ουρά[rear] ← νέο_στοιχείο
    ΑΛΛΙΩΣ
        ! σε κάθε άλλη περίπτωση αυξάνουμε τον
        δείκτη rear κατά 1
        rear ← rear + 1
        ! και εισάγουμε το στοιχείο
        ουρά[rear] ← νέο_στοιχείο
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

ΕΞΑΓΩΓΗ

```
! αρχικά ελέγχεται αν η ουρά είναι άδεια
ΑΝ (front= 0 ΚΑΙ rear=0) ΤΟΤΕ
    ! αν ναι τότε εμφανίζεται σχετικό μήνυμα
    ΓΡΑΨΕ 'η ουρά είναι άδεια'
    ! ελέγχεται αν η ουρά έχει ΜΟΝΟ ένα
    στοιχείο
    ΑΛΛΙΩΣ_ΑΝ (front = rear) ΤΟΤΕ
        ! οπότε εξαγουμε το στοιχείο της
        θέσης front
        ΓΡΑΨΕ ουρά[front]
        ! και μηδενίζουμε τους 2 δείκτες
        front ← 0
        rear ← 0
    ΑΛΛΙΩΣ
        ! σε κάθε άλλη περίπτωση εξαγουμε το
        στοιχείο της θέσης front
        ΓΡΑΨΕ ουρά[front]
        ! και αυξάνουμε τον front κατά 1
        front ← front + 1
    ΤΕΛΟΣ_ΑΝ
```

Υλοποίηση της ουράς με την βοήθεια πίνακα

Μια ουρά μπορεί να υλοποιηθεί με τη βοήθεια ενός **μονοδιάστατου** πίνακα και **δύο δεικτών** (των *front* και *rear*).

Για την εισαγωγή ενός στοιχείου στην ουρά αυξάνεται ο δείκτης *rear* κατά ένα και στη θέση αυτή αποθηκεύεται το στοιχείο.

Αντίστοιχα για τη λειτουργία της εξαγωγής, εξέρχεται το στοιχείο που δείχνει ο δείκτης *front*, ο οποίος στη συνέχεια αυξάνεται κατά ένα, για να δείχνει το επόμενο στοιχείο που πρόκειται να εξαχθεί.

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.

Παρατηρήσεις

Στη δομή δεδομένων **ουρά** παρατηρούμε τα εξής:

1. Η μέθοδος επεξεργασίας της στοίβας είναι η **FIFO** (First In First Out)
2. Κάθε νέο στοιχείο προστίθεται στο **τέλος** της ουράς.
3. Κάθε φορά μπορούμε να επεξεργαστούμε **μόνο** το στοιχείο που βρίσκεται στην αρχή της ουράς.
4. Για να επεξεργαστούμε ένα στοιχείο που **δεν** βρίσκεται στην αρχή της ουράς πρέπει να **εξαγάγουμε** όλα τα στοιχεία που βρίσκονται μπροστά του.
5. Μπορεί να γίνει **εξαγωγή**, από την **αρχή** της ουράς, μόνο **ενός** στοιχείου κάθε φορά.

Κατηφόρης