

מבנה נתונים מופשט – ADT – Abstract Data Type

הגדרה: מבנה נתונים המאופיין ע"י הפונקציות המופעלות עליו, כלומר אין התייחסות לצורת המימוש שלו ואיך הנתונים נשמרים בזיכרון אלא רק צורת השימוש במבנה, איזה פעולות ניתן להפעיל עליו.

מחסנית – Stack

מבנה: Last In First Out - LIFO - נכנס אחרון יוצא ראשון.

הפונקציות:

הפעולה	שם הפונקציה	זמן ריצה
הכנסה לראש המחסנית	Push()	O(1)
הוצאה מראש המחסנית והחזרת הערך	Pop()	O(1)
הצצה לראש המחסנית	Top()	O(1)
האם המחסנית ריקה	Empty()	O(1)

מימוש הפונקציות:

באמצעות מערך:

(עם אינדקס עזר שמתקדם עם כל הוספה ויורד עם כל הסרה)

Boolean Empty(S)

```
if S.numElements = 0
    return True
else
    return False
```

Key Pop(S)

```
if Empty(S)
    error "underflow"
else S.numElements ← S.numElements - 1
    return S[S.numElements + 1]
```

Push (S, Key)

```
S.numElements ← S.numElements + 1
S[S.numElements] ← Key
```

Key Top()

```
return S[S.numElements]
```

באמצעות רשימה מקושרת:

(הוספה והסרה מתבצעים בראש הרשימה)

Boolean Empty(S)

```
return head = nil
```

Key Top()

```
if head ≠ nil
    return head.key
else
    return error "isEmpty"
```

Key Pop(S)

```
if Empty(S)
    error "underflow"
else
    node ← head
    head ← node.next
    node.next ← nil
    return node.key
```

Push (S, Key)

```
node ← new node
node.key ← key
node.next ← head
head ← node
```

תור – Queue

מבנה: First In First Out - FIFO - נכנס ראשון יוצא ראשון.

הפונקציות:

הפעולה	שם הפונקציה	זמן ריצה
הכנסה לסוף התור	Enqueue()	$O(1)$
הוצאה מראש התור והחזרת הערך	Dequeue()	$O(1)$
האם התור ריק	Empty()	$O(1)$

מימוש הפונקציות:

באמצעות מערך:

(עם 2 אינדקסים read-שומר את האיבר הראשון שנכנס, ו-write – שומר אינדקס לאיבר האחרון שנכנס).

Enqueue(Q, x)

```
if n = capacity
    error "overflow"
Q[write] ← x
write ← (write + 1) mod capacity
n ← n + 1
```

Empty(Q):

```
return n = 0
```

Dequeue(Q):

```
if Empty(Q)
    error "underflow"
x ← Q[read]
read ← (read + 1) mod capacity
n ← n - 1
return x
```

באמצעות רשימה מקושרת:

(הוספה מתבצעת לסוף הרשימה והסרה מראש הרשימה)

Enqueue: use List.PushBack

Dequeue: use List.TopFront and List.PopFront

Empty: use List.Empty

לסיכום:

- תור ומחסנית הם מבני נתונים מופשטים
- ניתן לממש מחסנית ותור באמצעות מערך ורשימה מקושרת
- כל הפעולות על מחסנית ותור- זמן הריצה שלהן הוא $O(1)$