

## מבנה נתונים מופשט – ADT

הגדירה : מבנה נתונים המאפשר ע"י הפעולות המופיעות עליו, כלומר אין התייחסות לצורת המימוש שלו ואיך הנתונים נשמרים בזיכרון אלא רק צורת השימוש במבנה, איזה פעולות ניתן להפעיל עליו.

### **מחסנית – Stack**

**מבנה :** מבנה – Last In First Out - LIFO .

**הfonקציות:**

זמן ריצה	שם הfonקציה	הפעולה
O(1)	Push()	הכנסה לראש הממחסנית
O(1)	Pop()	הוצאה מראש הממחסנית וחזרת הערך
O(1)	Top()	הצאה לראש הממחסנית
O(1)	Empty()	האם הממחסנית ריקה

**שימוש הfonקציות:**

**באמצעות מערך :**

(עם אינדקס עוז שמתקדם עם כל הוספה ו יורד עם כל הסרת)

#### **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 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
```

#### **Key Top()**

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

## תור – Queue

**מבנה:** Queue – First In First Out - FIFO

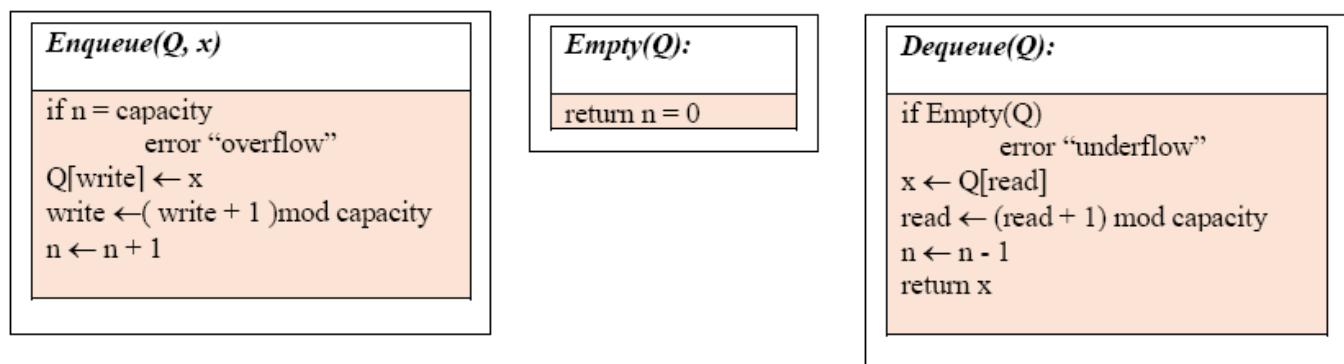
**הפונקציות:**

זמן ריצה	שם הפונקציה	הפעולה
O(1)	Enqueue()	הכנסה לסוף התור
O(1)	Dequeue()	הוצאת ראש התור והחזרת הערך
O(1)	Empty()	האם התור ריק

**מימוש הפונקציות:**

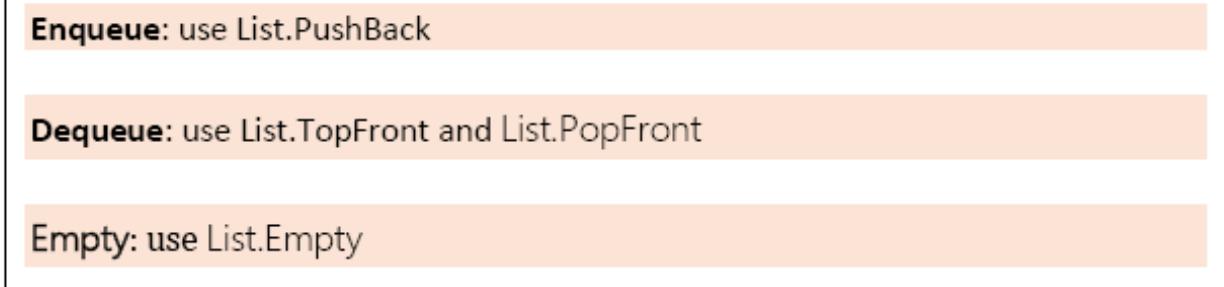
באמצעות מערך:

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



באמצעות רשימה מקוורת:

(הוספה מתבצעת לסוף הרשימה והסרה מראש הרשימה)



**לסיכום:**

- תור ומחסנית הם מבני נתונים מופשטים
- ניתן למשוחש ממחסנית ותור באמצעות מערך ורשימה מקוורת
- כל הפעולות על מחסנית ותור – זמן הריצה שלן הוא O(1)