**מחסנית ותור**

**מחסנית**

מחסנית (Stack) היא סוג של אוסף התומך בפעולות הכנסה והוצאה, כך שמתקיימת התכונה הבאה: פעולת מחיקה מחזירה תמיד את הערך שהוכנס אחרון, מבין ערכים הקיימים במחסנית. פעולות במחסנית מתבצעות רק בקצה אחד הנקרא **ראש המחסנית**.

הפרוטוקול המגדיר את דרך הגישה לערכים במחסנית נקרא LIFO

(**L**ast **I**n **F**irst **O**ut).

**פעולות:**

1. create(S) – יצירת מחסנית ריקה S.
2. push(S,x) – הכנסת האיבר x לראש המחסנית S.
3. top() – מחזירה את האיבר שנמצא בראש המחסנית (המחסנית עצמה לא משתנה).
4. pop() – הוצאת איבר מראש המחסנית והחזרתו, אם מחסנית ריקה יוחזר NULL.
5. stack\_empty()– מחזירה true אם מחסנית S ריקה, אחרת מחזירה false.

**מימושים אפשריים:**

1. רשימה מקושרת חד כיוונית. הכנסת והוצאת האיבר שנמצא בראש הרשימה.
2. מערך בגודל N עם מצביע לראש המחסנית. לא מאפשר שמירה של יותר מ-N איברים. למערך – מחסנית יש תכונה (משתנה) top.

**פסאודו-קוד לכמה פונקציות (שימוש במערך)**

S.top = 0

stack\_empty(S)

if S.top = 0

return True

else

return False

push(S, x)

if S.top = N

throw error “overflow”

else

S.top <- S.top + 1

S[S.top] <- x

pop(S)

if stack\_empry(S)

throw error “underflow”

else

S.top <- S.top – 1

return S[S.top] + 1]

**שאלה 1**

מה מבצע קטע קוד הבא? מה סיבוכיות האלגוריתם?

F(n)

create(S)

while n > 0 do

push(S, n mod 2)

n <- n/2

while not stack\_empty(S) do

print(S.pop)

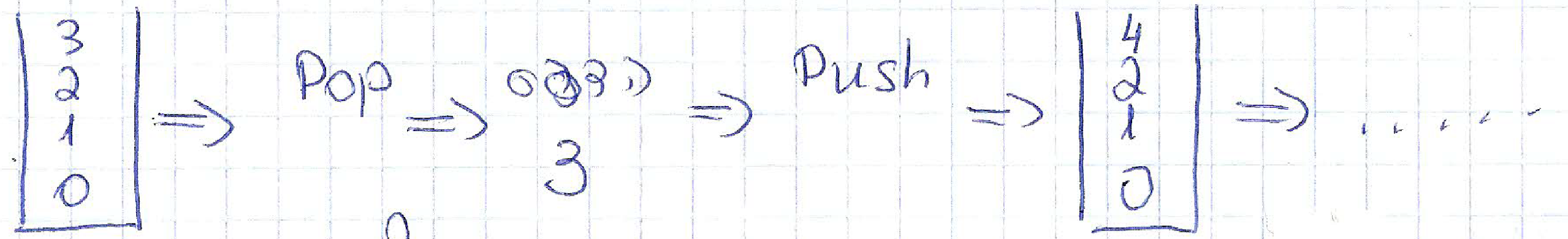
**שאלה 2**

פעולה Push מכניסה ערכים מ-0 עד 9 לפי סדר למחסנית.

פעולה Pop מוציאה את ערך מהמחסנית ומדפיסה אותו.

**דוגמה:**

אחרי 4 פעמים Push אזי המחסנית נראת



מה בין **הפלטים** הבאים (ערכים שהודפסו) אינו תקין בהנחה שבוצעו 10 פעולות PUSH ו-10 פעולות POP בסדר כלשהו?

1. 4, 3, 2, 1, 0, 9, 8, 7, 6, 5
2. 4, 6, 8, 7, 5, 3, 2, 9, 0, 1
3. 2, 5, 6, 7, 4, 8, 9, 3, 1, 0
4. 4, 3, 2, 1, 0, 5, 6, 7, 8, 9

**שאלה 3**

כתבו פסאודו-קוד להפיכת סדר הספרות של מספר עשרוני בעזרת מחסנית.

**לדוגמה:** 2047 🡨 7402

**תור**

תור (Queue) הוא מבנה נתונים שתומך בעיקרון FIFO (**F**irst **I**n **F**irst **O**ut), כלומר האיבר הראשון שנכנס לתור הוא האיבר הראשון שיצא ממנו.

**פעולות:**

1. **create(Q)** – יצירת תור ריק Q.
2. **head(Q)** – מחזיר את האיבר הראשון בראש התור (התור עצמו לא משתנה)
3. **enqueue (x)** – הכנסת איבר x לסוף תור Q.
4. **dequeue (Q)** – הוצאת איבר מראש התור Q.
5. **queue\_empty(Q)** – מחזיר True אם תור ריק, אחרת מחזיר False.
6. **queue\_full(Q)** – מחזיר True אם תור מלא ו-False אחרת.

**פסאודו-קוד לפעולות על תור במימוש על ידי מערך:**

|  |  |
| --- | --- |
| queue\_empty(Q)  if(Q.tail=Q.head)  return true  else  return false | queue\_full(Q)  if(Q.head=(Q.tail+1) mod n)  return true  else  return false |
| enqueue(Q , x)  if(Q.head=(Q.head+1) mod n)  throw error "overflow"  else  Q.tail <- Q.tail+1 mod n  Q[Q.head] <- x | dequeue(Q)  if(queue\_empty(Q))  Throw error "underflow"  else  Q.head <-(Q.head+1)mod n  return Q[Q.head+1] |

**מימושים אפשריים:**

1. רשימה מקושרת חד כיוונית. הכנסת האיברים מכיוון ה-"זנב" (tail) של רשימה והוצאת תעשה מכיוון ה-"ראש" (head).
2. מערך חד ממדי בגודל N עם שני אינדקסים.

**שאלה 4**

מה מבצעה פסאודו-קוד הבא? מה סיבוכיות של האלגוריתם?

F(Q)

create(S)

while not queue\_empty(Q) do

push(S, dequeue(Q))

while not stack\_empty(S) do

enqueue(Q,S.pop)

**שאלה 5**

כתוב פסאודו-קוד למימוש תור על ידי שתי מחסניות.

**אלגוריתם של פתרון:**

**מטרה של שאלה - לכתוב מחדש פונקציות Queue\_Empty, Enqueue ו-Dequeue**. באמצעות פונקציות של המחסנית.

נחזיק שתי מחסניות. נקרא להן inbox ו-outbox.

* כאשר נדחוף לתור – נדחוף תמיד ל-inbox.
* כאשר נשלף מהתור, נבצע את הפעולה הבאה:
  + אם יש ב-outbox איבר – נשלוף אותו.
  + אם outbox ריקה:
    - להעתיק כל איברים מ- inbox ל-outbox
    - נשלוף איבר מ-outbox.

**שאלה 6**

כתבו פעולה (פונקציה) המקבלת תור של מספרים שלמים ומחזירה את סכום האיברים הזוגיים הנמצאים בו.

אחרי הפעלת פונקציה, התור יישאר ללא מספרים זוגיים.

מה סיבוכיות של אלגוריתם?

**תרגילים נוספים**

**שאלה 1 מחסנית**

מה יהיה תוכן של המחסנית אחרי סדרת פעולות הבאה:

push(E)

push(A)

push(S)

pop()

push(Y)

pop()

push(Q)

push(U)

push(E)

pop()

pop()

pop()

push(S)

push(T)

pop()

pop()

pop()

push(I)

push(O)

pop()

push(N)

pop()

pop()

pop()

**שאלה 2 מחסנית**

מה משמעות של פונקציה? איזה ערך היא מחזירה?

**Mystery (int n){**

stack s ;

s.push(0);

s.push(1);

**for** (int i <- 0 to n-2)

x <- s.pop();

y <- s.pop();

s.push(x);

s.push(x + y);

return s.pop();

}

**שאלה 3 תורים**

נתונים שני תורים 2Q , 1Q. שני התורים מכילים מספרים שלמים חיוביים, וממוינים בסדר עולה כך שבראש התור נמצא המספר הקטן ביותר.  
כתבו אלגוריתם אשר יחזיר את המספר הקטן ביותר המופיע בשני התורים בו זמנית. אם אין מספר המופיע בשני התורים יחזיר האלגוריתם 1-.

דוגמא :   
3,7,13,20,47,70 ----> 1Q  
12,20,65,70,75 ----- > 2Q - על הפונקציה להחזיר 20.