

## מבני נתונים – תרגיל מס. 3

1. שלוש דרכים מקובלות לטיול על עץ בינארי הן pre-order, post-order ו-in-order. שמות השיטות נובעים מהמקום בו נמצא השורש ביחס לתתי-העצים שתחתיו. פסאודו קוד לשלושת השיטות הוא (הקריאה הראשונה היא כאשר  $v$  הוא השורש):

PRE\_ORDER(v)

```
IF v = NULL THEN RETURN
ELSE
    PRINT v.key
    PRE_ORDER(v.left)
    PRE_ORDER(v.right)
RETURN
```

POST\_ORDER(v)

```
IF v = NULL THEN RETURN
ELSE
    POST_ORDER(v.left)
    POST_ORDER(v.right)
    PRINT v.key
RETURN
```

IN\_ORDER(v)

```
IF v = NULL THEN RETURN
ELSE
    IN_ORDER(v.left)
    PRINT v.key
    IN_ORDER(v.right)
RETURN
```

- א. האם יתכן ש-pre-order של עץ מסוים יהיה זהה ל-in-order של אותו עץ? אם כן, איזה תנאי צריך העץ לקיים, אם לא הוכיחו.
- ב. האם יתכן ש-post-order של עץ מסוים יהיה זהה ל-in-order של אותו עץ? אם כן, איזה תנאי צריך העץ לקיים, אם לא הוכיחו.
- ג. האם יתכן ש-pre-order של עץ מסוים יהיה זהה ל-post-order של אותו עץ? אם כן, איזה תנאי צריך העץ לקיים, אם לא הוכיחו.

2. הציעו מימוש לא רקורסיבי לפרוצדורה IN\_ORDER שבשאלה הקודמת, בעל אותו חסם זמן ריצה אסימפטוטי.

3. הציעו אלגוריתם שמקבל כקלט עץ בינארי בעל  $n$  צמתים, ומחזיר כפלט את מספר הצמתים שמקיימים את התנאי הבא: המפתח של  $v$  גדול מכל המפתחות של הצאצאים שלו (ישירים ולא ישירים). זמן הריצה של האלגוריתמים צריך להיות  $O(n)$ .

4. נניח שאנחנו מנהלים בית מלאכה לאריזת מתנות בקופסות. יש לנו מלאי של קופסות קובייתיות, שבו לכל קופסה יש נפח ידוע שהיא יכולה להכיל. קופסות יכולות להתווסף או להיגרע מהמלאי. כשמגיעה בקשה לאריזת מתנה, הלקוח יודע את גודל הקופסה הקטן ביותר שמתאים למתנה, אבל אנחנו נספק לו את הקופסה הקטנה ביותר המתאימה מתוך המלאי שלנו. המשימה בשאלה זאת היא לתכנן מבנה נתונים לניהול מלאי הקופסות. על מבנה הנתונים לתמוך בפעולות הבאות:

- $InsertBox(size)$  – הוספת קופסה בגודל  $size$  למלאי.
  - $RemoveBox(size)$  – הוצאת קופסה בגודל  $size$  מהמלאי.
  - $GetBox(size)$  – החזרת גודל הקופסה הקטנה ביותר במלאי שגודלה הוא יותר מ- $size$ .
  - $ExistsBox(size)$  – בדיקה האם יש במלאי קופסה גדולה מ- $size$ .
- הערות: מתחילים ממבנה נתונים ריק. הפרמטר  $size$  הוא מספר שלם חיובי. יתכנו מספר קופסות באותו גודל. ההבדל בין  $GetBox$  ו- $ExistsBox$  היא שהראשונה מחזירה את גודל הקופסה שנרצה לספק ללקוח, בעוד השנייה רק בודקת האם קיימת קופסה מתאימה. יש להראות מימוש שבו זמן הריצה של כל פעולה הוא  $O(\log n)$ , וטוב ככל שתוכלו.

5. נגדיר מבנה נתונים שמכיל מספרים שלמים ותומך בפעולות  $Insert$ ,  $Delete$  ו- $Find$  כרגיל, ובנוסף בפעולת  $Pair(p)$  שמחזירה זוג איברים מהמבנה שסכומם הוא  $p$  בדיוק, אם יש זוג כזה.

א. הציגו מימוש בעל זמן ריצה טוב ביותר (כלומר זמן הריצה של הפעולה האיטית ביותר, הוא הנמוך ביותר שתוכלו להשיג).

ב. הציגו מימוש שבו הפעולה  $Pair$  לוקחת  $O(1)$  זמן (ושאר הפעולות עלולות לקחת זמן רב).

הערות:

- שימו לב שהעצים הבינאריים שמופיעים בשאלות הם לא בהכרח עצי חיפוש.