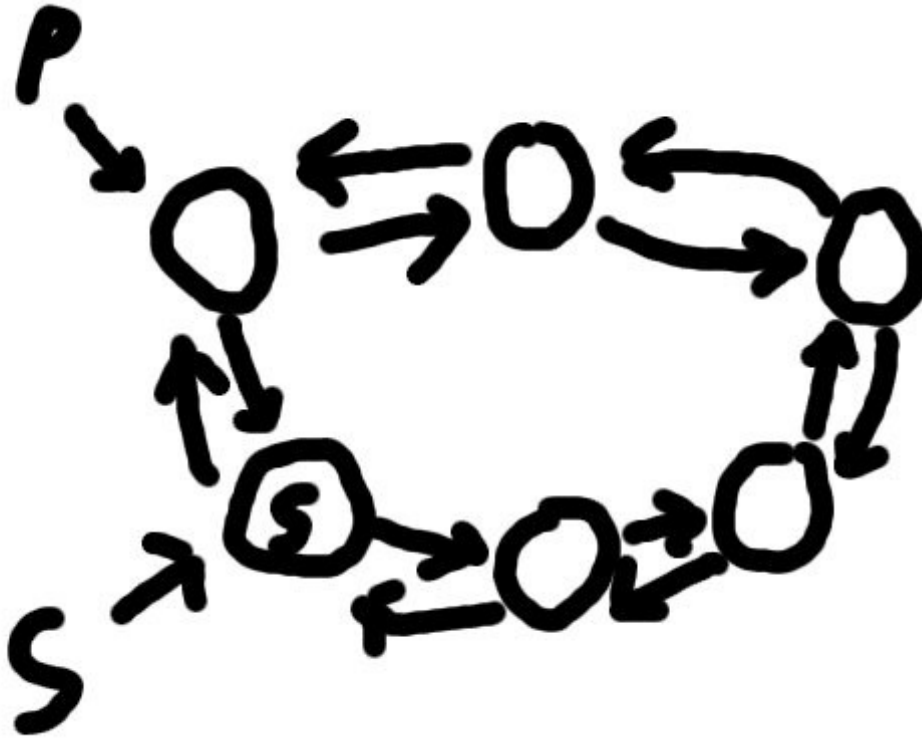


1. המשימה שהוטלה על מספר קבוצות של ילדים ומדריכיהם באחד השבטים בצופים היא ליצור שרשרת שתקיף את בניין השבט. לצורך מעקב אחר ביצוע המשימה פותחה תוכנה מיוחדת התומכת בפעולות הבאות: (מבנה הנתונים יהיה רשימה דו-מקושרת מעגלית)



INIT(S) - אתחול מבנה הנתונים. בהתחלה השרשרת ריקה.
 S יצביע לאובייקט סנטינל (אובייקט המייצג את סוף ותחילת הרשימה) שמצביע לעצמו (איתחולו יהיה בתוך אותה שגרה) (סיבוכיות $O(1)$)
 INSERT-KID(S, p, id) - הכנסת רשומה של ילד נוסף שמס' תעודת הזהות שלו הוא id למבנה S,
 יבנה אובייקט חדש שמייצג את התלמיד בעל תעודת הזהות, ימתחו מצביעים בין האובייקט ל P ולבין הבא אחרי P בהתאם. (סיבוכיות $O(1)$)
 INSERT-ADULT(S, p, id) - הכנסת רשומה של מדריך שמס' תעודת הזהות שלו הוא id למבנה S, מיד לאחר הרשומה של הילד/המדריך ש-p מצביע עליו.
 יבנה אובייקט חדש שמייצג מבוגר בעל תעודת הזהות המצויינת, ימתחו מצביעים בין האובייקט ל P ולבין הבא אחרי P בהתאם, (סיבוכיות $O(1)$)
 REMOVE(S, p) - הוצאת הרשומה ש-p מצביע עליה מהמבנה S.
 מתיחת המצביעים שלהאובייקט לפני האובייקט שאליו מצביע P לאובייקט שאחריו (סיבוכיות $O(1)$)
 CLOSEST-ADULT(S, p) - החזרת מס' תעודת הזהות של המדריך הקרוב ביותר בשרשרת לילד ש-p מצביע עליו.
 השמת מצביע PRE לאובייקט שלפני P ומצביע POST לאובייקט שאחרי P,
 בדיקה אם אחד מהם מצביע למדריך (PRE או POST) במידה וכן יוחזר מספר תעודת הזהות, במידה ולא פעם אחת יקודם PRE ופעם אחת POST עד למציאת המבוגר הקרוב ביותר או עד פגישתם. (סיבוכיות $O(N)$)

כתוב שגרה להדפסת עץ חיפוש בינארי איטרטיבית בעזרת מחסנית בסיבוכיות $O(n)$

```
printTree(t,stack)
  insert(stack,t)
  while notEmpty(stack) do
    remove(stack)
  insert(stack,t)
  while t != null do
    push(stack,t)
    t ← t.left
  remov(stack)
  x ← stack.pop
  if right(x) not nil
    insert(stack,x)
```

3. בשאלה זו עומדות לרשותך שתי מחסניות ועליך לממש באמצעותן תור בצורה יעילה.

כלומר, יש לבצע את הפעולות הבסיסיות המוגדרות על תור

באמצעות ביצוע פעולות מתאימות על שתי המחסניות (ENQUEUE,DEQUEUE)

(a) הסבר בקצרה איך יתבצעו הפעולות השונות על התור, נסמן ב- n את מספר האיברים הנמצאים בזמן נתון בתור. מה תהיה סיבוכיות הזמן של כל אחת מהפעולות המבוצעות על התור?

תשובה: הפעולות השונות שיתבצעו על התור יהיו enqueue הכנסת איבר למחסנית א' שתייצג את הזנב של התור, ו dequeue הוצאת איבר ממחסנית ב' שתייצג את ראש התור, הפעולה dequeue תדרוש שהמחסנית ב' הלאו היא מחסנית הזנב תהיה מלאה, במידה ואינה מלאה נבצע שפיכה של מחסנית א לתוכה, סיבוכיות זמן הריצה של הכנסה תהיה $O(1)$ כמובן, וסיבוכיות של הוצאה תהיה $O(1)$ בממוצא מאחר וגם במקרה הגרוע רק הוצאה של אחד מהאיברים תיקח $O(n)$ וכל ההוצאות האחרות כתוצאה מכך יהיו ב $O(1)$

(b) כתוב שגרות למימוש הפעולות enqueue ו dequeue:

```
enqueue(x)
  tailStack.push(x)
dequeue()
  if headStack.isEmpty?
    Then until tailStack.isEmpty?
      Do headStack.push(tail.pop)
  headStack.pop
```

4. k

(a)

הראה את ריצת אלגוריתם הגיבוב הכפול על סדרת הקלט הבאה, במערך בגודל 19:
4,76,23,9,56,4,78,5,3,4,65,7,39,54,87,1,8,56

שלבים:

```
[]
[nil, nil, nil, nil, 4]
[76, nil, nil, nil, 4]
[76, nil, nil, nil, 4, nil, nil, nil, nil, nil, nil, 23]
[76, nil, nil, nil, 4, nil, nil, nil, nil, 9, nil, 23]
[76, nil, nil, nil, 4, nil, nil, nil, nil, 9, nil, 23, nil, nil, nil, nil, nil, 56]
[76, nil, nil, nil, 4, nil, nil, nil, nil, 9, nil, 23, nil, nil, 4, nil, nil, nil, 56]
[76, nil, 78, nil, 4, nil, nil, nil, nil, 9, nil, 23, nil, nil, 4, nil, nil, nil, 56]
```

[76, nil, 78, nil, 4, 5, nil, nil, nil, 9, nil, 23, nil, nil, 4, nil, nil, nil, 56]
 [76, nil, 78, 3, 4, 5, nil, nil, nil, 9, nil, 23, nil, nil, 4, nil, nil, nil, 56]
 [76, nil, 78, 3, 4, 5, nil, nil, nil, 9, 4, 23, nil, nil, 4, nil, nil, nil, 56]
 [76, nil, 78, 3, 4, 5, nil, nil, 65, 9, 4, 23, nil, nil, 4, nil, nil, nil, 56]
 [76, nil, 78, 3, 4, 5, nil, 7, 65, 9, 4, 23, nil, nil, 4, nil, nil, nil, 56]
 [76, 39, 78, 3, 4, 5, nil, 7, 65, 9, 4, 23, nil, nil, 4, nil, nil, nil, 56]
 [76, 39, 78, 3, 4, 5, nil, 7, 65, 9, 4, 23, nil, nil, 4, nil, 54, nil, 56]
 [76, 39, 78, 3, 4, 5, nil, 7, 65, 9, 4, 23, nil, nil, 4, nil, 54, 87, 56]
 [76, 39, 78, 3, 4, 5, nil, 7, 65, 9, 4, 23, nil, 1, 4, nil, 54, 87, 56]
 [76, 39, 78, 3, 4, 5, 8, 7, 65, 9, 4, 23, nil, 1, 4, nil, 54, 87, 56] במבנה הסופי

(b) צייר את טבלת הגיבוב המתקבלת מאותה סדרת קלט כאשר פותרים את בעיית ההתנגשויות על ידי שרשור.

[[76], [39, 1], [78], [3], [4, 23, 4, 4], [5], nil, [7], [65, 8], [9], nil, [87], nil, nil, nil, nil, [54], nil, [56]]
 כל תת מערך מייצג רשימה מקושרת המתחילה משמאל לימין (לדוגמא במקום 4 במערך הנ"ל נמצאת רשימה מקושרת המכילה את המספרים 4,23,4,4 כאשר 23 הוא המספר השני ברשימה).

(c) עבור איזה מהמבנים שיצרת, המבנה מסעיף א' או המבנה מסעיף ב', מספר הפעולות הנדרשות לענות על השאלה: "האם 23 בטבלא?" יהיה מינימלי? עבור איזה מהם מספר הפעולות הנדרשות לענות על השאלה: "האם המספר 29 נמצא בטבלא?" - יהיה מינימלי

• המבנה לו ידרש מספר הפעולות המינימלי לענות על השאלה "האם 23 בטבלא?" הוא המבנה מסעיף ב'

• המבנה לו ידרש מספר הפעולות המינימלי לענות על השאלה "האם 29 בטבלא?" הוא המבנה מסעיף ב'

5. נתון מערך A בגודל N המכיל שורש N ערכים שונים מתוך התחום [k..1]

כאשר K קבוע, המטרה היא למיין את עברי המערך, לצורך כך, הצע מבנה נתונים

ואלגוריתם המשתמש בו, זמן הריצה הנדרש הוא $O(n)$ במקרה הממוצע.

הערה: מבנה הנתונים יכול להיות מורכב ממספר מבני נתונים בסיסיים.

תשובה: אשתמש במערך עזר של רשימות מקושרות ואמיין במיין בסיס,

וכך הסיבוכיות הממוצעת תהיה $O(N)$

שם: יעקב קילימניק

ת"ז: 300446408

מנחה: דביר לנצברג

מספר מטלה: 15