**פתרון מבחן במבנה נתונים תש"פ**

1. נשלוף את כל האיברים ממחסנית B לטבלת גיבוב. [O(n) הכנסות בO(1) בממוצע - > סה"כ O(n)]

וכעת נתחיל בהדפסה:

נעבור על כל איבר במחסנית A ונשלוף אותו, [O(1)] נבדוק אם הוא קיים ב טבלת הגיבוב [ממוצע O(1)], אם לא נדפיס אותו.

סה"כ עבור כל המחסנית A O(n)

וממילא זמן הריצה הממוצע:

O(n)+O(n)=O(n)



**int complete(bt t**)

**{**

**if (t==Null)**

**return -1;**

**int l=complete(left(t));**

**int r=complete(right(t));**

**if (l==r && l!=-2)**

**return l+1;**

**else return -2;**

**}**

3. לכל משפט כתבי אם הוא נכון או לא נכון. נמקי בקצרה.

א.      כל אלגוריתם לחיפוש איבר x בערמה, רץ, במקרה הגרוע, בזמן (n)Θ לפחות.

נכון, מכיוון שעל n/2 **העלים** בעץ ודאי שאין לנו שום מידע מי גדול ממי ממילא חייבים לעבור על כולם במקרה הגרוע והרי (n)Θ =  (n/2)Θ

ב.      סיבוכיות זמן הריצה לשיעורין תמיד קטנה מסיבוכיות  זמן הריצה במקרה הגרוע ביותר.

לא נכון, ישנם מצבים בהם זמן הריצה אינו משתנה בין מקרה למקרה, ואז ודאי אין הבדל.

לדוג': ביצוע n פעמים של מיון מיזוג.

ג.       אם מבצעים רוטציה על צמת בעץ AVL, מספר העלים בעץ לפני הרוטציה שווה למספר העלים בעץ אחרי הרוטציה.

לא נכון, לדוג':

לאחר גלגולים:

4.

הרעיון הכללי של הפיתרון:

נשמור משתנה המכיל את כמות האיברים במבנה ונאתחל ב-0 – מכונה count

נכנה את שלוש המחסניות A, B, C

בפעולת הכנסה : נכניס למחסנית A ונעלה את count ב1

בremoveLast אם count>0 נוציא איבר מהמחסנית A ונוריד את count ב1

בremoveFirst נפעל כך:

אם Count>1

אם המחסנית B ריקה, נעבור ונשלוף את כל איברי המחסנית A וכל איבר נכניס לB וכן העתק שלו נכניס ל C [בסוף התהליך B וC יהיו זהות– ויכילו את האיבר הראשון שהיה בA בראש המחסנית]

לאחר מכן נשלוף מC את כל האיברים ונחזיר לA חוץ מאת האיבר התחתון בC[כך שA תכיל את הנתונים המקוריים בדיוק באותו סדר חוץ מהאיבר שיצא]

ונשלוף מB את ראש המחסנית ונחזיר בסוף בפונקציה [כך שזה יהיה האיבר הראשון שנכנס] וכן נוריד את count ב1

אם המחסנית B אינה ריקה נשלוף את ראש המחסנית ונחזיר בסוף הפונקציה ונוריד את count ב1

ניתוח סיבוכיות:

במקרה שB ריקה O(n)

במקרה שהיא מלאה O1)

זמן ממוצע בשיטת הבנקאי: נאמר שכל הכנסה עולה 3 פעולות נוספות ממה שהיא עולה בפועל [5 במקום 3 – בכל מקרה עדיין O(1)] ואז כשאר נגיע לפעולה יקרה – העתקת A לB ולC והחזרת C לA , יהיהו לנו מספיק "מטבעות" ביד לביצוע הפעולה.

וממילא הזמן הממוצע O(1).

isEmpty() בודק האם count=0

5.

המשתנה last צריך להיות גלובלי / סטטי/ להיות פרמטר בפונקציה

node last=null;

node search(node root, node p)

{

if(root==null) return "not found"

if(root==p) return last;

node p1 =search(root.left);

if(p1!="not found") return p1;

last=root;

return search(root.right);

}