Data Structures Final Exam – 11.2.2020 – with Solutions

שאלה 1 (20 נקודות)

Describe an algorithm that gets two stacks, A and B, and prints all the items in A that do not appear in B.

The average running time of the algorithm should be O(n) (n is the number of items in both stacks.)

You need not write code, only a description of the algorithm.

תארי אלגוריתם המקבל שתי מחסניות, A ו-B, ומדפיס את האיברים שנמצאים רק ב-A ולא ב-B.

זמן הריצה של האלגוריתם O(n) בממוצע (n הוא מספר האיברים בשתי המחסניות).

שימי לב: בשאלה זו אין צורך לכתוב קוד. מספיק לתאר את האלגוריתם במילים.

נכניס את איברי מחסנית B לטבלת גיבוב (hash table). נעבור על אברי A: כל איבר נחפש בטבלת הגיבוב עם אותה פונקצית גיבוב, ואם הוא לא נמצא נדפיס אותו

שאלה 2 (16 נקודות)

A "triangle" binary tree is an empty binary tree or a binary tree of height h (h≥0) with 2^{h+1}-1 nodes. In other words: an empty binary tree or a binary tree of height 0 with one node, or a binary tree of height 1 with 3 nodes, or a binary tree of height 2 with 7 nodes etc.

Complete the following function (in C) that receives a pointer to a binary tree (type bt. Assume it is implemented. you do not have to implement it) and returns the height of the tree (the height of an empty tree is -1) if the tree is a "triangle" tree, and returns -2 otherwise. Assume the functions left and right were implemented, and return the left child and the right child of a node.

The running time of the algorithm should be O(n). n is the number of nodes in the tree.

```
int complete(bt t)
{
 if (t==Null)
      __________

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

 if (l==r && _____)
```

```
      return ____
 else return -2;
}
```

עץ בינרי "משולש" הוא עץ ריק או עץ בינרי בגובה h (h≥0) בעל $2^{h+1}-1$ צמתים. כלומר: עץ ריק או עץ בגובה 0 בעל צמת יחיד, או עץ בגובה 1 בעל 3 צמתים, או עץ בגובה 2 בעל 7 צמתים וכו'.

השלימי את הפונקציה הבאה (ב-C) שמקבלת מצביע לעץ בינרי (טיפוס bt. הניחו שמימושו נתון. אין צורך לממשו) ומחזירה את גובה העץ (גובהו של עץ ריק הוא 1-) אם העץ "משולש", ו 2- אחרת. הניחי שהפונקציות left ו-right מומשו והן מחזירות את הבן השמאלי והימני של t, בהתאמה.
זמן הריצה של האלגוריתם O(n) (n הוא מספר הצמתים בעץ).

```
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
}
```

בשתי ההשלמות האחרונות אפשר להחליף את l ב-r (כי הם שווים).

שאלה 3 (22 נקודות)

For every sentence write False or True. Explain your answer.

a. Every algorithm for searching a given x in a heap, will run, in the worst case, in time Θ(n), at least.
b. The amortized running time is always smaller than the worst-case running time.
c. If a rotation is performed on a node in an AVL tree, the number of leaves before the rotation will be equal to the number of leaves after the rotation.

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

א. כל אלגוריתם לחיפוש איבר x בערמה, רץ, במקרה הגרוע, בזמן Θ(n) לפחות.
ב. סיבוכיות זמן הריצה לשיעורין תמיד קטנה מסיבוכיות זמן הריצה במקרה הגרוע ביותר.
ג. אם מבצעים רוטציה על צמת בעץ AVL, מספר העלים בעץ לפני הרוטציה שווה למספר העלים בעץ אחרי הרוטציה.

א. נכון. אם, למשל, האיבר שמחפשים בערמת מקסימום קטן מכל האיברים בערמה, על האלגוריתם להשוות אותו לכל העלים בערמה. אחרת יכול להיות שהוא שווה לעלה אליו הוא לא הושווה.
ב. לא נכון. למשל, אם נבחר פונקציית פוטנציאל שהיא זהותית 0, זמן הריצה לשיעורין וזמן הריצה במקרה הגרוע יהין שוויו.
ג. לא נכון. אם לשורש יש שני בנים (שני עלים) ונבצע רוטציה על השורש, נקבל עץ עם עלה יחיד.

שאלה 4 (22 נקודות)

Implement, using three stacks and O(1) extra memory, a data structure with the following operations:

a. insert(ds,x) – inserts item x to the data structure ds.
b. removeLast(dx) – removes from ds, and returns, the last item to be inserted to the data structure.
c. removeFirst(dx) – removes from ds, and returns, the first item to be inserted to the data structure (from the items currently in ds).
d. isEmpty(ds) – returns true if the data structure is empty and false otherwise.

The amortized running should be O(1), for every operation.

Describe your implementation and how the operations are performed. Show that the running time is as required.

עליך לממש בעזרת שלוש מחסניות ו-O(1) זיכרון נוסף, מבנה נתונים עליו מוגדרות הפעולות הבאות:

א. insert(ds, x) – מכניסה את x למבנה ds.
ב. removeLast(ds) – מוחקת מהמבנה ומחזירה את האיבר האחרון שהוכנס למבנה
ג. removeFirst(ds) - מוחקת מהמבנה ומחזירה את האיבר הראשון שהוכנס למבנה, מבין האיברים שבמבנה.
ד. isEmpty(ds) – מחזירה true אם מבנה הנתונים ריקה ו-false אחרת.

זמן הריצה לשיעורין של כל פעולה צריך להיות O(1).

תארי את מבנה הנתונים ואת אופן הביצוע של כל פעולה. הראי שזמן הריצה של המימוש שלך עומד בתנאי השאלה.

המבנה יורכב ממחסנית הראש, מחסנית הזנב, מחסנית עזר ושני שדות נוספים (מונים) שיאחסנו את מספר האיברים במחסנית הראש ובמחסנית הזנב.

א. הכנסה תתבצע תמיד למחסנית הראש. המונה של מספר האיברים במחסנית הזנב יגדל ב-1.
ב. אם במבנה איבר יחיד נחזיר אותו ונאפס את המונה.
אם מחסנית הראש ריקה אז נעביר את מחצית האיברים ממחסנית הזנב למחסנית העזר (אם גם מחסנית הזנב ריקה אז יש ניסיון שליפה ממבנה ריק), נעביר את האיברים שנותרו המחסנית הזנב למחסנית הראש ונחזיר את האיברים ממחסנית העזר למחסנית הזנב.

נעדכן את המונים של שתי המחסניות. הוצאת האיבר שנכנס אחרון תתבצע ממחסנית הראש והמונה יקטן ב-1.

ג. אם במבנה איבר יחיד נחזיר אותו ונאפס את המונה.
אם מחסנית הזנב ריקה אז נעביר את מחצית האיברים ממחסנית הראש למחסנית העזר (אם גם מחסנית הראש ריקה אז יש ניסיון שליפה ממבנה ריק), נעביר את האיברים שנותרו במחסנית הראש למחסנית הזנב ונחזיר את האיברים ממחסנית העזר למחסנית הראש.
נעדכן את המונים של שתי המחסניות. הוצאת האיבר שנכנס ראשון תתבצע ממחסנית הזנב והמונה יקטן ב-1

ד. אם מספר האיברים 0 יוחזר TRUE, אחרת יוחזר FALSE

נגדיר פונקציית פוטנציאל שהיא:3 כפול הפרש מספר האיברים בין שתי המחסניות (בערך מוחלט).

בהכנסה – העלות האמתית היא O(1). השינוי בפוטנציאל הוא הגדלה ב-3 או הקטנה ב-3. סה"כ העלות לשיעורין היא קבועה.

בהוצאה (גם מהראש וגם מהזנב) – במקרה הטוב העלות האמתית O(1) והשינוי בפוטנציאל הוא הגדלה ב-3 או הקטנה ב-3. במקרה הגרוע העלות האמתית היא 3n כי סה"כ 3 חצאי מחסניות מועברות (וכל ההעברה היא 2 פעולות: PUSH ו-POP). הפוטנציאל קטן ב-3n והעלות לשיעורין היא קבועה.

העלות של בדיקה האם המבנה ריק היא O(1) ואין שינוי בפוטנציאל.

שאלה 5 (20 נקודות)

Write, in pseudocode, an algorithm that receives a pointer p to a node in a binary tree and returns a pointer to the previous node in an inorder scan. If p is the first pointer in the inorder scan the algorithm will return null.

The running time of the algorithm should be O(h). h is the height of the tree.

The algorithm must contain no more than 12 lines (each line will contain only one instruction). Do not forget to ident your pseudocode.

כתבי, בפסידוקוד, אלגוריתם המקבל מצביע p לצמת בעץ בינרי, ומחזיר מצביע לצמת הקודם ל-p בסדר תוכי. אם p הוא ראשון בסדר תוכי יוחזר null.

זמן הריצה של אלגוריתם יהיה O(h), כש-h הוא גובה העץ.

האלגוריתם יכיל 12 שורות לכל היותר. כל שורה תכיל פקודה אחת בלבד. אל תשכחי לעמד את הקוד ולבצע הזחות, כנדרש.

```
prev(p)
if left(p)≠null
then p←left(p)
     while right(p)≠null do
       p←right(p)
     return p
while p≠null and left(parent(p))=p do
      p←parent(p)
return p
```