

מבני נתונים תרגיל בית 3

תאריך פרסום 30.4.17

הגשה 16.5.17 23.59

מרצה ומתרגלים אחראים: קלרה קדם, ניצן פרחי, דן שולמן

הנחיות:

הגשת העבודה הינה בזוגות •. העבודה חייבת להיות מוקלדת •. יש להגיש קובץ בפורמט pdf למערכת ההגשה

•. שאלות לגבי העבודה יש לשאול בפורום באתר הקורס או בשעות קבלה של המרצה/המתרגל האחראיים על העבודה .

שאלה 1:

הוכיחו או הפריכו את הטענות הבאות: תנו הסבר קצר לתשובותיכם.

א. אם T עץ AVL בגובה h , אז כל הרמות עד הרמה ה- $h-2$ (כולל) מלאות.

הגדרה: הרמה i של עץ היא אוסף כל הצמתים בעומק i .

נאמר שהרמה i מלאה אם מספר האיברים בה הוא 2^i .

למשל: רמת השורש היא רמת האפס ובה $2^0 = 1$ איברים כאשר היא מלאה.

ב. פעולת המחיקה מעץ חיפוש בינארי היא חלופית (כלומר, העץ המתקבל ממחיקת x

ולאחריו y זהה לעץ המתקבל ממחיקת y ולאחריו x , לכל x, y בעץ).

ג. המספר המינימאלי של צמתים בעץ AVL בגובה $h=4$ הוא 12.

ד. נתון עץ חיפוש בינארי T בעל n צמתים. רוצים לבדוק האם T מקיים את תכונת האיזון

של עץ AVL. אזי זמן הריצה של האלגוריתם היעיל ביותר לבדיקה הנ"ל במקרה הגרוע

הוא $\Theta(\log n)$

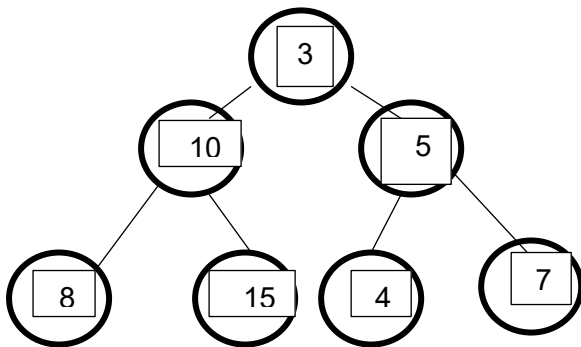
שאלה 2:

"עץ חיפוש מקולקל קלות" הוא עץ בינארי שבו בדיוק צומת אחד לא מקיים את תכונת החיפוש. במילים אחרות, קיים צומת יחיד, v , עם מפתח $v.key$, כך שמתקיים בדיוק אחד מהתנאים הבאים:

- בתת העץ השמאלי של v קיים לפחות מפתח אחד שגדול מ- $v.key$.
 - או
 - בתת העץ הימני של v קיים לפחות מפתח אחד שקטן מ- $v.key$.
- צומת v כזה ייקרא צומת מקולקל.

- א. תארו אלגוריתם למציאת צומת מקולקל בעץ
- ב. בסעיף זה נרצה להפוך עץ חיפוש מקולקל קלות לעץ חיפוש.
- הציעו אלגוריתם, יעיל ככל האפשר, המתקן עץ חיפוש מקולקל קלות (הופך אותו לעץ חיפוש חוקי בעל אותם מפתחות) מבלי לשנות את המבנה (צורה/שלד) שלו.

דוגמא לעץ מקולקל קלות



שאלה 3

נתון תור Q עם n מספרים. תארו אלגוריתם הממיין את המספרים בתור מהמספר הקטן לגדול. ניתן להשתמש רק בתור עזר ובמספר קבוע של משתני עזר. נתחו את סיבוכיות זמן הריצה של האלגוריתם שתיארתם.

שימו לב: בעבודה עם התור אפשר להשתמש אך ורק בפעולות שמוגדרות על התור:

`isEmpty`, `enqueue`, `dequeue`

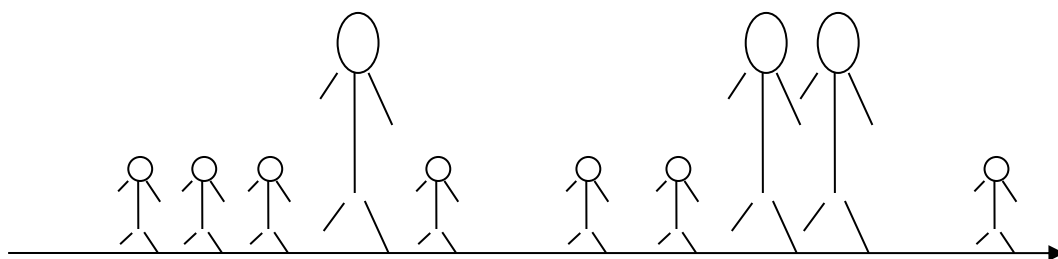
הפתרון צריך להכיל הסבר מילולי לאלגוריתם המלווה בפסאודו-קוד.

שאלה 4

הראו שניתן להפוך עץ חיפוש בינארי נתון כלשהו, בעל n צמתים לעץ חיפוש בינארי אחר שצורתו נתונה, תוך שימוש ב $O(n)$ רוטציות. (רמז: תחילה הראו שנחוצות לכל היותר $n-1$ רוטציות כדי להפוך את העץ לשרשרת ימנית).

שאלה 5

על כוכב הלכת "גוליבר" חיים להם ביחד הגמדים והענקים. הגמדים אוהבים מאוד לדבר, גמד ידבר עם כל גמד אחר שהוא רואה. לעומתם הענקים שונאים רעש! עליכם לתחזק מבנה נתונים עבור הגמדים והענקים שעומדים אחד לצד השני על קו ישר.



על המבנה לתמוך בפעולות הבאות

שם פעולה	תאור פעולה	זמן ריצה דרוש
<i>Init()</i>	אתחול מבנה נתונים ריק.	$O(1)$ במקרה הגרוע
<i>InsertDwarf(location)</i>	הכנסת גמד במיקום <i>location</i> (המיקום הוא מספר ממשי המייצג נקודה על ציר ה- X . המיקום הוא המזהה של הגמד ואינו משתנה לאורך זמן). ניתן להניח שלא קיים יצור במיקום <i>location</i> במבנה לפני פעולת ההכנסה.	$O(\log n)$ במקרה הגרוע
<i>InsertGiant(location)</i>	הכנסת ענק במיקום <i>location</i> . (המיקום הוא מספר ממשי המייצג נקודה על ציר ה- X . המיקום הוא המזהה של הענק ואינו משתנה לאורך זמן). ניתן להניח שלא קיים יצור במיקום <i>location</i> במבנה לפני פעולת ההכנסה.	$O(\log n)$ במקרה הגרוע

$O(\log n)$ במקרה הגרוע	<p>החזרת תשובה לשאלה "האם הגמד במיקום L1 רואה את הגמד במיקום L2?" (ולכן גם מדבר איתו).</p> <p>שני גמדים רואים זה את זה אם אין ענק ביניהם.</p> <p>במידה ולא קיים גמד באחד המיקומים L1 או L2 יש להחזיר הודעת שגיאה.</p>	<i>IsTalking(L1,L2)</i>
$O(\log n)$ במקרה הגרוע	<p>מחק את היצור (גמד או ענק) שנמצא במיקום location. אם אין יצור במיקום location יש להדפיס הודעת שגיאה.</p>	<i>Remove(location)</i>
$O(\log n+k)$ במקרה הגרוע, כאשר k מספר הגמדים שיכולים לדבר עם הגמד הנמצא במיקום location	<p>יש להדפיס את מיקומי כל הגמדים שיכולים לדבר עם הגמד הנמצא במיקום location. ניתן להניח שקיים גמד במיקום location ואין צורך לבדוק זאת.</p>	<i>WhomTalking(location)</i>

הבהרה: n – הינו מספר היצורים במבנה ברגע נתון.

תארו בקצרה את מבנה הנתונים, ספקו אלגוריתם לכל אחת מהשיטות והוכיחו עמידה בזמני הריצה.