

[דף סיכום בחינה](#)

מספר שאלה	ניקוד מירבי	ציון
1	30.00	30.00

ציון בחינה סופי : 30.00

הבחינה הבדוקה בעמודים הבאים

שאלה 1

Type	Usage	Space – Complexity
<i>Degue < Object[] ></i>	אחסון מערך בגודל 3 בתור דו כיווני : קודקוד שיצא מן האיזון, קודקוד ההכנסה וסוג הרוטציה שהתבצעה	$O(1)$
<i>boolean</i>	תמיכה באחסון של מערך בתור דו כיווני : בודקת האם התבצעה רוטציה בהכנסה	$\Theta(1)$
<i>Node</i>	תמיכה באחסון של מערך בתור דו כיווני : שומר את קודקוד ההכנסה	$\Theta(1)$
<i>int</i>	תמיכה בפעולות <i>Select</i> ו- <i>Rank</i> : שומר את מספר הקודקודים בתת העץ	$\Theta(1)$

...

שאלה 2

Operation	Number – Of – Repetitions	Total – Time – Complexity
הוצאת מערך מתור דו כיווני והגדרת משתנים	1	$\Theta(1)$
<i>node.decrease()</i> פונקציית עזר - עדכון שדה <i>size</i> בקודקודים	1	$O(h)$
<i>rotation(node, ImbalanceCases.RR)</i> פונקציית עזר - ביצוע רוטציה <i>RR</i>	לכל היותר 1	$\Theta(1)$
<i>rotation(node, ImbalanceCases.LL)</i> פונקציית עזר - ביצוע רוטציה <i>LL</i>	לכל היותר 1	$\Theta(1)$
ניתוק קודקוד ההכנסה מההורה	1	$\Theta(1)$
<i>node.updateHeight()</i> לולאה - עדכון שדה <i>height</i> בקודקודים	לכל היותר <i>h</i>	$O(h)$

...

שאלה 3

Type	Usage	Space – Complexity
<i>Degue < Object[] ></i>	אחסון מערך בגודל 4 בתור דו כיווני : קודקוד לפני הפיצול \ קודקוד ההכנסה הערך האמצעי \ ערך ההכנסה סוג הפעולה הנדרשת - פיצול \ מחיקה האינדקס של הערך האמצעי בקודקוד הנוכחי	$O(h)$

...

שאלה 4

Operation	Number – Of – Repetitions	Total – Time – Complexity
הוצאת מערך מתור דו כיווני והגדרת משתנים	לכל היותר $1 + h$	$O(h)$
$node.removeKey(value)$ מחיקת מפתח מהקודקוד לפי ערך	לכל היותר $1 + h$	$O((2t - 1) \cdot h)$
$node.removeChild(index)$ מחיקת ילד מהקודקוד לפי אינדקס	לכל היותר $2h$	$O(2t \cdot h)$
$node.addChild(node);$ הוספת ילד לקודקוד	לכל היותר h	$O(h \cdot n \cdot \log(n))$

...

שאלה 5

זמן הריצה לפעולת *backtrack* לפי ההצעה של דני הוא $\Theta(1)$.

בפעולת *insert*, שומרים את המצביע לקודקוד של תת העץ לפני ההכנסה.

ובפעולת *backtrack*, מעדכנים את הקודקוד המתאים במצביע שאחסנו.

...

שאלה 6

הפתרון של דני יעיל יותר מבחינת סיבוכיות זמן ריצה מהפתרון שלנו אך אינו יעיל יותר מבחינת סיבוכיות מקום.

עבור כל פעולת *backtrack* הפתרון של דני דורש אחסון זיכרון כגודל העץ ואילו הפתרון שלנו דורש אחסון זיכרון כגובה העץ.