<u>חלק תיאורטי- תומר רודניצקי (20662787820) ועידו רוזינר (209617000)</u>

:AVLNode להלן תיעוד

פונקציה	תיאור	חישוב סיבוכיו
		ת זמן
getLeft()	הפונקציה מחזירה את הבן השמאלי של הצומת. מוגדר	0(1)
	להיות None במקרה והצומת וירטואלית.	
getRight()	הפונקציה מחזירה את הבן הימני של הצומת. מוגדר	0(1)
	להיות None במקרה והצומת וירטואלית.	
getParent()	הפונקציה מחזירה את ההורה של הצומת. מוגדר להיות	0(1)
	None במקרה והצומת וירטואלית	
getValue()	הפונקציה מחזירה את הערך של הצומת. מוגדר להיות	0(1)
	None במקרה והצומת וירטואלית.	
getHeight()	הפונקציה מחזירה את הגובה של הצומת. מוגדר להיות	O(1)
	1- במקרה והצומת וירטואלית.	
getSize()	הפונקציה מחזירה את הגודל של הצומת. מוגדר להיות 0	O(1)
	במקרה והצומת וירטואלית.	
getBF()	הפונקציה מחזירה את ה-balance factor של הצומת.	0(1)
setLeft(node)	הפונקציה מקבעת את הבן השמאלי של הצומת להיות	0(1)
	.node	
setRight(node)	הפונקציה מקבעת את הבן הימני של הצומת להיות	O(1)
	.node	
setParent(node)	הפונקציה מקבעת את ההורה של הצומת להיות node.	0(1)
setValue(value)	הפונקציה מקבעת את הערך של הצומת להיות value.	0(1)
setHeight(h)	הפונקציה מקבעת את הגובה של הצומת להיות h.	0(1)
setSize(s)	הפונקציה מקבעת את הגודל של הצומת להיות s.	0(1)
setBF(bf)	של הצומת balance factor-הפונקציה מקבעת את ה	0(1)
	להיות bf.	
isRealNode()	אם הצומת היא לא צומת True הפונקציה מחזירה	0(1)
	וירטואלית (הגובה שונה מ-(1-)) ואחרת False.	
computeBF()	של balance factor-הפונקציה מחשבת ומחזירה את	0(1)
	הצומת. כלומר, מחשבת ומחזירה	
	1 + height(leftSubtree) - (1 + height(rightSubtree))	
computeHeight	הפונקציה מחשבת ומחזירה את הגובה של הצומת.	0(1)
()	כלומר, מחשבת ומחזירה	
	1 + max{height(leftSubtree), height(rightSubtree)	
computeSize()	הפונקציה מחשבת ומחזירה את הגודל של הצומת.	0(1)
35p3.00.23()	יופונון או בווסבונ ובווריו אוניותוי סידיובוונוונו כלומר, מחשבת ומחזירה את מספר הצמתים שקטנים שווים לה –	· (-)
	1 + size(leftSubtree) + size(rightSubtree)	

:AVLTreeList להלן תיעוד

פונקציה	תיאור	חישוב סיבוכיות זמן
getRoot()	הפונקציה מחזירה את	0(1)
	השורש של העץ. מוחזר	
	AVLNode(None)	
	במקרה והעץ וירטואלי.	
getMax()	הפונקציה מחזירה את	0(1)
	הצומת המקסימלית בעץ.	
	מוחזר None במקרה	
math dia ()	והעץ וירטואלי	0(1)
getMin()	הפונקציה מחזירה את	0(1)
	הצומת המינימלית בעץ.	
	מוחזר None במקרה והעץ וירטואלי.	
getSize()	ווזעץ ויו טוארי. הפונקציה מחזירה את	0(1)
get3ize()	וופונקציון מווויו וו אונ גודל העץ - מספר	0(1)
	גורי דוען - מספר הצמתים הלא וירטואליים	
	שנמצאים בעץ. מוחזר 0	
	במקרה והעץ וירטואלי.	
setRoot(node)	הפונקציה מקבעת את	0(1)
	השורש של העץ להיות	0 (1)
	.node	
setMax(node)	הפונקציה מקבעת את	0(1)
,	הצומת המקסימלית בעץ	
	להיות node.	
setMin(node)	הפונקציה מקבעת את	0(1)
	הצומת המינימלית בעץ	
	להיות node.	
setSize(s)	הפונקציה מקבעת את	0(1)
	הגודל של העץ להיות s.	
empty()	הפונקציה מחזירה True	0(1)
	אם הרשימה (העץ) ריקה	
	עץ וירטואלי) ואחרת	
	.False	
retrieve(i)	הפונקציה מחזירה את	הפונקציה קוראת ל -
	i-ערך האיבר במקום ה	treeSelect(self, i+1), שרצה ב-
	אם קיים, אחרת היא מחזירה None.	וכל שאר הפעולות, $O(\log i)$
	נווויו וו וויסווו.	קורות ב- $0(1)$. לכן הפונקציה
incort/i vol\	בפינדעוב מבנוסב עובב	רצה ב-O(log i).
insert(i, val)	הפונקציה מכניסה איבר בעל ערך val לרשימה	תחילה הפונקציה קוראת ל- treeSelect(i) שרצה בסיבוכיות
		· · ·
	במקום ה-i במידה וקיימים לפחותi איברים ברשימה.	$O(\log i)$. לאחר מכן הפונקציה קוראת ל-
	הפונקציה מחזירה את	יאווו מכן חפונקציו קוו אוניי- rebalanceTree(node) שרצה
	וופונקציון מוווין וו אונ מספר פעולות האיזון	בסיבוכיות $O(\log n)$. מכיוון שכל
	מספר פעהיות רואייון שנדרשו בשלב תיקון העץ	שאר הפעולות קורות ב- $O(\log n)$
	על מנת לשמר את תכונת	פאר וופעה ווניקוו ווניבי(ד) ס הפונקציה רצה בסיבוכיות
	עז מנונ ז סמו אונ ונכונונ האיזון.	$O(\log n)$
		$O(\log n)$

תחילה הפונקציה קוראת ל- (igi) אחד הפונקציה אם הוא קיים. (indecity) אוד הפונקציה קוראת ל- (igi) אחד השלח במיבוכיות (igi) (ig			
או לפונקציה (2010) או לפונקציה ברשימה, אם הוא קיים. (2010) או לפונקציה בסיבוכיות (2010, 2010) אחר מכן הפונקציה קוראת ל- (2010) אחר מכן הפונקציה קוראת ל- (2010) אחר השני. מכיוון שכל שאר שרצות בסיבוכיות (2010, 2010) אחר השני. מכיוון שכל שאר הפונקציה קוראת ל- (2010, 2010) אחר השני. מכיוון שכל שאר הפונקציה קוראת ל- (2010, 2010) אחר הפונקציה קוראת לפונקציה הפונקציה קוראת ל- (2010, 2010) אחר הפונקציה מודרה את או (2010) אחר הפונקציה מודרה את או (2010) אחר הפונקציה קוראת ל- (2010, 2010) אחר הפונקציה אור (2010, 2010) אחר הפונקציה מודרה את אור הפונקציה קוראת ל- (2010, 2010) אחר איים ב- (2010, 2010) אחר הפונקציה קוראת ל- (2010, 2010) אחר הפונקציה קוראת ל- (2010, 2010) אחר הפונקציה קוראת ל- (2010, 2010) אחר איים ב- (2010, 2010) אחר איים ב- (2010, 2010) אחר איים ב- (2010, 2010) אחר אורים ב- (2010, 2010) אחר איים ב- (2010, 2010) א	delete(i)	הפונקציה מוחקת את	
א מונר שרצה בסיבוכיות (מ מונר מונר מינר מונר מינר מינר מינר מינר מינר מינר מינר מי		i-האיבר במקום ה	שרצה בסיבוכיות treeSelect(i)
לחר מון	1	ברשימה, אם הוא קיים.	או לפונקציה $O(\log i)$
לאחר מכן הפונקציה קוראת ל- delete_node(node_to_del) delete_node(node_to_del) delete_node(node_to_del) (rebalanceTree(node) (rebalanceTree(node) (rebalanceTree(node) (rebalanceTree(node)), what had be part a corici (m (n		הפונקציה מחזירה את	שרצה בסיבוכיות successor(x)
לאחר מכן הפונקציה קוראת ל- delete_node(node_to_del) delete_node(node_to_del) delete_node(node_to_del) (rebalanceTree(node) (rebalanceTree(phon) (repalanceTree(phon) (repalanceTree(phon)) (repalanc		מספר פעולות האיזוו	$O(\log n)$
לל מנת לשמר את תכונת ול - (delete_node(node_to_del), ארו ולי (compal) איזון. של איזון. שר שרצות בסיבוכיות (mapl) (maple) (ma	1	· •	_
רצות בסיבוכיות (n			
שרצות בסיבוכיות (מ gol) שרצות ברים ברשימה הפונקציה קורת ב-(1) של שאר הפונקציה מוחקת את הפונקציה מוחקת את הפונקציה קוראת ליש בסיבוכיות (מ gol) של פעולת האיזון של העץ. הצומת שממנה נתחיל את המעולת ב-(1) של המשול הפונקציה קוראת ליש בסיבוכיות (מ gol) של לל לכל היותר ב-(1) של פעולת האיזון של העץ. הצומת שממנה נתחיל את הפעולות בפונקציה קורות ב-(1) של הפונקציה קורות ב-(1) של הפונקציה קורות ב-(1) של הפונקציה קורות ב-(1) של הפונקציה קוראת לפונקציה שממנה ריקה. ברשימה היקה. בסיבוכיות (מ gol) של השבל את הערכים של הפונקציה קוראת לפונקציה השבל השבל השבל אוברי הרשימה (העץ) לפי ברשימה היקה. הפעולות קורות ב-(1) של שאר הערכים של הפונקציה הערך של הפונקציה מנוית (מ gol) של איברי הרשימה (העץ) לפי הרשימה הערך של הפונקציה מנוית (מ gol) של איברי הרשימה (העץ) לפי הרשימה הערך של הפונקציה מנוים את הערך של הפונקציה מנוים את הערך של הפונקציה קוראת ליש (מ gol) של איברי הרשימה (העץ) לפי הרשימה (העץ) לפי הרשימה הערך של הפונקציה מנוים את הערך של הפונקציה מנוים את הערך של הפונקציה קוראת ליש (a gol) של איברי הרשימה (העץ) של איברי הרשימה (העץ) לפי הרשימה (העץ) לפי הרשימה (העץ) לפי הרשימה הערך של הפונקציה מוליריתם הבים בים בים בים בים בים בים בים בים בים			
אחרי השני. מכיוון שכל שאר הפעולות קורות ב-(10). delete_node(node_to_del) הפונקציה רצה בסיבוכיות (20 m) node_to_del node_to_del node_to_del node_to_del node_to_del node_to_del neuror aceius (20 m). מבינוי (20 m) node_to_del node_to_del node_to_del node_to_del neuror neuror node_to_del neuror neuror node_to_del neuror node_to_del neuror neuror node_to_del neuror neuror node_to_del neuror neu	1	•	, ,
הפעולות קורות ב- (1) 0, הפונקציה רצה בסיבוכיות הפונקציה קוראת לפונקציה האושלות בפוניות ((1) 0 פערים. מכיוון שכל שאר ((1) 0 הפונקציה קוראת לפונקציה הישימה ((1) 0 הפונקציה קוראת לפונקציה הישימה ((1) 0 ברשימה, או ending sin and sin a	1		, , , ,
פונקציה רצה בסיבוכיות הסופר בעל שני ילדים הפונקציה קורות ב- (100 equingrangle and prode to del mode (node to del mode (node) (pode) אמיתיים הפונקציה קוראת ל- (100 equingrangle and product and produ	1		•
delete_node(node_to_del) An all of the product	1	וופונוןב וו נוווו ווויי.	
שו node_to_del אמרתיים הפונקציה מוחקת את מרולים הפונקציה קוראת ל- מערך האיבה השולות בפונקציה קוראת לפונקציה המידירה את הפונקציה קוראת לפונקציה המידירה מערך ברשימה היקה. ברשימה העץ) לפי ברשימה העץ או מערך היק אם הפעולות קורות ב-(1)0, מכיוון שכל שאר הפונקציה מניסים הפעולות קורות ב-(1)0, מכיוון שכל שאר הפונקציה המידירה עושה העץ) לפי הרשימה העץ) לפי הרשימה העץ) לפי סדר האינדקסים (חום הפונקציה מודירה את לפונקציה מודירה את לפי סדר האינדקסים. הפונקציה מודירה את לפי סדר האינדקסים. הפונקציה מודירה את לפי סדר האינדקסים. הפונקציה קוראת לפונקציה הפונקציה הפונקציה קוראת לפונקציה מספר האיברים ברשימה (חיל). מספר האיברים ברשימה (חילה הפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה קוראת לפונקציה הפונקציה הפונקציה הפונקציה קוראת לפונקציה הפונקציה הפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה הפונקציה הפונקציה הפונקציה הפונקציה הפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה קוראת לפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה ל			•
אמיתיים הפוקציה קוראת ל- בסיבוכיות ((or n) (or noin) לכל היותר ((or noin)) לכל הפונקציה קוראת לפונקציה ((or noin)) לכל של היבר הרשימה (העץ) לפי הרפינה ((or noin)) לרשימה ביקה ((or noin)) לרשימה ((or n			
או פעולת האיזון של שאר (listToArray) (מרום של מושל של האיזון של האיזון של האיזון של האיזון של העץ. (מרום של שאר הפונקציה קורות ב- (מרום של מרום של	delete_node(node_to_del)		
בסיבוכיות $(n \log p)$, לכל היותר 0 (log n) פעמים. מכיוון שכל שאר 0 (nog n) פעולת בפונקציה קורות ב- 0 (log n) הפעולות בפונקציה קורות ב- 0 (log n) הפונקציה רצה בסיבוכיות 0 (log n) (log n) הפונקציה רצה בסיבוכיות הפונקציה קוראת לפונקציה ברשימה, או None ברשימה, או None ברשימה, או None ברשימה היקה. בסישומה הפונקציה קוראת לפונקציה ברשימה (העץ) (וווער ב- 0) הפונקציה קוראת לפונקציה הפונקציה מחזירה את הפעולות קורות ב- 0 (listToArray) הפונקציה מביבוכיות 0 (0). מכיוון שכל שאר בסיבוכיות 0 (0) מכיוון שכל שאר הפונקציה רצה בסיבוכיות 0 (0). הפונקציה מביבוכיות 0 (0) הפונקציה מביבוכיות 0 (0) הפונקציה מוסים של איברי הרשימה (העץ) לפי סדר האינדקסים. של איברי הרשימה (העץ) הפונקציה קוראת לשל הפונקציה קוראת ל- 0 (listToArray) הפונקציה מחזירה את (0 (0) הפונקציה מחזירה את (0) הפונקציה קוראת ל- 0 (0) הפונקציה מחזירה את (0) הפונקציה קוראת ל- 0 (0) הפונקציה מחזירה את (0) הפונקציה קוראת ל- 0 (0) הפונקציה מחזירה את (0) הפונקציה קוראת ל- 0 (0) הפונקציה מחזירה את (0) הפונקציה קוראת ל- 0 (0) הפונקציה מחזירה את (0) הפונקציה קוראת ל- 0 (0) הפונקציה מחזירה את (0) (0) הפונקציה קוראת ל- 0 (0) הפונקציה מחזירה את (0) (0) הפונקציה קוראת ל- 0 0 (0) הודר הישימה בסדר (0 0	1		
פעמים. מכיוון שכל שאר (10 הפונקציה קורות ב- (10 הפונקציה קורות ב- (10 הפונקציה רצה בסיבוכיות (10 הפונקציה מחזירה את (10 הפונקציה מחזירה את (10 הפונקציה קוראת לפונקציה מחזירה את (10 הפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה קוראת לפונקציה קוראת לפונקציה קוראת לפונקציה מודירה את הפונקציה רצה בסיבוכיות (10 מודיר לפיסדר האינדקסים (10 מודיר לפיסדר מודיר לפיסדר האינדקסים (10 מודיר לפיסדר האינדקסים (10 מודיר לפיסדר האינדקסים (10 מודיר לפיסדר האינדקסים (10 מודיר לפיסדר האינדים מודיר לפיסדר האינדקסים (10 מודיר לפיסדר האינדקסים (10 מודיר לפיסדר האינדקסים (10 מודיר לפיסדר האינדקסים (10 מודיר לפיסדר האינדים מודיר לפיסדר הראינדים מודיר לפיסדר הראשור לפיסדר הראשור לפיסדר האינדים מודיר לפיסדר הראשור לפיסדר לפיסדר הראשור לפיסדר הראשור לפיסדר הראשו		•	
הפעולות בפונקציה קורות ב- $O(\log n)$ ($O(\log n)$	1		` ~ /
(10) הפונקציה רצה בסיבוכיות (10) הפונקציה מחזירה את (10 הפונקציה לרשימה, או None ברשימה, או None ברשימה, או None ברשימה היקה. הפונקציה מרוון שכל שאר (10). מרוון שכל שאר (10). הפעולות קורות ב-(10). הפעולות קורות ב-(10). הפעולות קורות ב-(10). הפעולות קורות ב-(10). הפונקציה מרניסה הפונקציה בשיעור (במקום ההכנסה מבוצעת כמו באלגוריתם הפונקציה מרניסה הפונקציה הערך של הפונקציה השימה (העץ) לפי הדפסה נוסיף את הערך של הפונקציה עושה זאת על הפונקציה הערך של הפונקציה מחזירה את (10). הפונקציה קוראת ל- הפונקציה קוראת ל- הפונקציה מחזירה את (10). הפונקציה מחזירה את (10). הפונקציה מחזירה את (10). הפונקציה קוראת ל- הפונקציה קוראת ל- הפונקציה קוראת ל- הפונקציה מחזירה את (10). הפונקציה מחזירה את (10). הפונקציה מחזירה את (10). הפונקציה קוראת ל- הפונקציה מחזירה את (10).		פעולת האיזון של העץ.	
(log n) $O(1)$ הפונקציה מחזירה את $O(1)$ ברשימה, או None ברשימה ריקה. ברשימה ריקה. ברשימה ריקה. ברשימה ריקה. ברשימה ריקה. ברשימה ריקה. ברשימה הירכים של הפונקציה קוראת לפונקציה מחזירה מערך האיבר האחרון הפעולות קורות ב- $O(1)$. ברשימה ריקה הפעולות קורות ב- $O(1)$, מכיוון שכל שאר הפנוקציה ריקה הפעולות קורות ב- $O(1)$, הפנוקציה ריקה הפעולות קורות ב- $O(1)$, הפונקציה ריקה. ברשימה ריקה הוסדירות של של הידי היד ביינות הערף של הידי היד היד היד היד היד היד היד היד הי			הפעולות בפונקציה קורות ב-
(1) הפונקציה מחזירה את (1) אורך האיבר הראשון (1970 ברשימה, או None ברשימה, או None ברשימה ריקה. (1) הפונקציה מחזירה את (1970 ברשימה ליקה. ברשימה לפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה הפונקציה מחזירה מערך (1970 מכיוון שכל שאר הפינות (1971 מיוון שכל שאר הפונקציה רצה בסיבוכיות (1971 מרשימה (העץ) לפי הפונקציה רצה בסיבוכיות (1971 מרשימה היקה. הוסיפה מבוצעת כמו באלגוריתם הפונקציה מכניסה הפונקציה מערך של הפינוסיף את הערך של הפונקציה מוסיף את הערך של הפונקציה מוסים. הפונקציה מוסים ולוסים אורכול מספר האיברים ברשימה. (1971 מספר האיברים בסדר. (1971 מספר משבר. (1971 מספר משבר. (1971 מספר משבר. (1971 מספר. (1			הפונקציה רצה בסיבוכיות $ hinspace hinspace 0(1)$
או א			$O(\log n)$
או א	first()	הפונקציה מחזירה את	0(1)
וdast() וברשימה ריקה. ומst() וברשימה ריקה. ומרן האיבר האחרון ומרן האיבר האחרון ומרן האיבר האחרון ברשימה או None ברשימה ריקה. ברשימה ריקה. ברשימה ריקה. ברשימה ומעץ לפי ומרן חושרצה הפונקציה מחזירה מערך בסיבוכיות (מושל שאר הפונקציה רצה בסיבוכיות (מושל בשיעור (במקום ההכנסה מבוצעת כמו באלגוריתם הדפסה נוסיף את הערך של הדפסה נוסיף את הערך של הפונקציה עושה זאת על הפונקציה עושה זאת על הפונקציה ומדירה את וואר וואר הפונקציה קוראת ל- ומספר האיברים ברשימה. ווstToArray(self) מרוילה הפונקציה קוראת ל- ומר למערך אוו ברוים ברשימה. ומספר האיברים ברשימה. ווstToArray(self) מרוילה הפונקציה קוראת ל- ומרן מוויים את הערף של הפונקציה מחדירת את ווואר ווואר ווויים ברשימה. מוויים אוואר איברי הרשימה בסדר וווויים אוואר איברי הרשימה בסדר ווווויים וווויים אווויים ברשימה וווויים אווויים ברשימה בסדר ווווויים ברשימה וווויים אווויים וווויים אוויים ברשימה בסדר ווווויים אוויים ברשימה בסדר וווויים ברשימה בסדר וווויים אוויים ברשימה ביים ברשימה בסדר וווויים אוויים ברשימה בסדר וווויים אוויים ברשימה בסדר וווויים ברשימה בסדר וווויים אוויים ברשימה בסדר וווויים אוויים ברשימה בסדר וווויים ברשימה בסדר וווויים ברשימה בסדר וווויים ברשימה בסדר וווויים ברשימה שוויים ברשימה בסדר וווויים ברשימה בסדר וווויים ברשימה בסדר ווויים ברשימה שוויים ברשימה בסדר ווויים ברשימה שוויים ברשימה בסדר ווויים ברשימה בסדר בסדר בסדר ברשימה בסדר ברשימה בסדר ברשימה בסדר בסדר ברשימה אוויים ברשימה ברשים ברשימה ברשים ברשים ברשים ברשימה ברשים ב	,	· ·	
ברשימה ריקה. last() הפונקציה מחזירה את None ערך האיבר האחרון ערך האיבר האחרון ברשימה ריקה. ברשימה ריקה. ברשימה לשגליה של שאר (moder (node, L) (moder (node)		-	
0(1) הפונקציה מחזירה את None ערך האיבר האחרון ברשימה, או None ברשימה, או nelitgran לפונקציה קוראת לפונקציה הפונקציה קוראת לפונקציה מסוזירה מערך המכיל את הערכים של inorder(node, L) מכיוון שכל שאר הפעולות קורות ב- $0(1)$ 0, מכיוון שכל שאר הפונקציה רצה בסיבוכיות $0(1)$ 1, מכיוון שכל שאר הפנוקציה רצה בסיבוכיות $0(1)$ 1, או מערך ריק אם הפונקציה רצה בסיבוכיות $0(1)$ 2, הפונקציה מכניסה מבוצעת כמו באלגוריתם הדפסה נוסיף את הערך של הדפסה נוסיף את הערך של הפונקציה עושה זאת על הפונקציה הפונקציה עושה זאת על הפונקציה הפונקציה מסדרת את $0(1)$ 1, הפונקציה מסדרת את $0(1)$ 2, הפונקציה מסדרת את $0(1)$ 3, הפונקציה קוראת ל-הפונקציה קוראת ל-הפונקציה קוראת ל-הפונקציה קוראת ל-הפונקציה קוראת ל-הונקציה מונקציה קוראת ל-הונקציה קוראת ל-הונקציה הונקציה מונקציה מונק			
ערך האיבר האחרון ריידים אומר אומר אומר אומר אומר איבר האחרון ריידים אומר אומר אומר אומר איברי הרשימה (העץ) לפי הפעולות קורות ב-(1)0, הפונקציה רצה בסיבוכיות (1)0, ההכנסה מבוצעת כמו באלגוריתם הפונקציה מכניסה הדפסה נוסיף את הערך של האיברי הרשימה (העץ) הדפסה נוסיף את הערך של הפונקציה הפונקציה עושה זאת על הפונקציה הפונקציה עושה זאת על הפונקציה (1)0. רצה בסיבוכיות (1)0, הפונקציה עושה זאת על הפונקציה מודירה את (1)0. הפונקציה מחזירה את (1)0. הפונקציה מחזירה את (1)0. מספר האיברים ברשימה. (1)0 הפונקציה קוראת ל- הפונקציה מסדרת את (1)10. איברי הרשימה בסדר (1)10.	last()		0(1)
אונדר אונדער אונדער בסיבוניות ($O(n)$) ולכן הפונקציה עובדער בסיבוניות ($O(n)$) ולכן הפונקציה אונדער בסיבוניות ($O(n)$) ולכן הפונקציה בשיעור (במקום הדפסה נוסיף את הערך של הדומת למערך ($O(n)$) ולכן הפונקציה של איברי הרשימה (העץ) ולכן הפונקציה לפי סדר האינדקסים. הפונקציה עושה זאת על הפונקציה ($O(n)$) ולכן הפונקציה של איברי הרשימה (העץ) ולכן הפונקציה של איברי הרשימה ($O(n)$) ולכן הפונקציה של איברי הרשימה ($O(n)$) ולכן הפונקציה מסובר האינדקסים. הפונקציה עושה זאת על איברי העץ. הפונקציה מחזירה את ($O(n)$) הפונקציה מחזירה את ($O(n)$) ולכן הפונקציה הפונקציה מחזירה את ($O(n)$) ולכן הפונקציה איברי העץ. הפונקציה מחזירה את ($O(n)$) ולכן הפונקציה איברי העץ. הפונקציה קוראת ל- הפונקציה מסדרת את ($O(n)$) ולכן הפונקציה איברי הרשימה בסדר ($O(n)$) ולכן הפונקציה איברי הרשימה ($O(n)$) ולכן הפונקציה איברי הרשימה ($O(n)$) ולכן הפונקציה איברי הרשימה ($O(n)$) הפונקציה מחזירה את ($O(n)$) ולכן הפונקציה איברי הרשימה ($O(n)$) הודער ($O(n)$) הפונקציה קוראת ל- הפונקציה מסדרת את ($O(n)$) ולכן הפונקציה קוראת ל- הפונקציה מחזירה את ($O(n)$) הידער הרשימה החידיר הרשימה רייק אונדער ל- הרשימה רייק איברי הרשימה רייק אונדער ל- הרשימה רייק איברי הרשימה רייק אים הייר הרשימה רייק איברי הרשימה רייק איברי הרשימה רייק איברי הרשימה רייק אים הייר הרשימה רייק אים הייר הרשימה רייק איברי הרשימה רייק איברי הרשימה רייק איברי הרשימה ריים איברי הרשימה ריים איברי הרשימה ריים איברי הרשימה ווידע הייר הריים איברי הרשימה הריים איברי הרשים הייר הייר היים איברי הרשימה הריים		•	- ()
ברשימה ריקה. הפונקציה קוראת לפונקציה מחזירה מערך (IistToArray() הפונקציה מחזירה מערך המכיל את הערכים של המכיל את הערכים של הפעולות קורות ב-(0,1), מכיוון שכל שאר הפונקציה רצה בסיבוכיות (0,1), מכיוון שכל שאר (order הפונקציה רצה בסיבוכיות (0,1), הפונקציה מכניסה הפונקציה רצה בסיבוכיות (1,1), הפונקציה מכניסה ההכנסה מבוצעת כמו באלגוריתם החדפסה נוסיף את הערך של הדפסה נוסיף את הערך של הפונקציה עושה זאת על הפונקציה (0,1), הפונקציה עושה זאת על הפונקציה (0,1), הפונקציה עושה זאת על הפונקציה קוראת ל- הפונקציה מסדרת את (1,1), הפונקציה קוראת ל- הפונקציה מדרת את (1,1),		I = = = = = = = = = = = = = = = = = = =	
הפונקציה קוראת לפונקציה החוזירה מערך וnorder(node, L) איברי הרשימה (העץ) לפי בסיבוכיות (n) . מכיוון שכל שאר הפונקציה רצה בסיבוכיות (n) . מכיוון שכל שאר הפונקציה רצה בסיבוכיות (n) . הפונקציה רצה בסיבוכיות (n) . הפונקציה מכניסה הפונקציה מבוצעת כמו באלגוריתם הדפסה נוסיף את הערך של הדפסה נוסיף את הערך של הצומת למערך (n) . הפונקציה עושה זאת על הפונקציה הפונקציה עושה זאת על הפונקציה קוראת ל- הפונקציה מסדרת את הפונקציה קוראת ל- הפונקציה מסדרת את (isstToArray). שיברי הרשימה בסדר (isstToArray). הפונקציה מסדרת את (inorder (node, L) הפונקציה מסדרת את (inorder (node, L) הפונקציה קוראת ל- הפונקציה מסדרת את (inorder (node, L) האיברי הרשימה בסדר (inorder (node, L) האיברי הרשימה בסדר (inorder (node, L) האיברי הרשימה בסדר (inorder (node, L) הפונקציה מסדרת את (inorder (node, L) הפונקציה מסדרת את (inorder (node, L) האיברי הרשימה בסדר (inorder (node, L) היברים של העץ או מערך (חוד ביצה בסיבוכיות (חוד ביצה בסיבוכיות (חוד ביצה מסדרת את (inorder (node, L) האיברי הרשימה (חוד ביצה בסיבוכיות (חוד ביצה ביצה ביצה (חוד ביצה ביצה ביצה (חוד ביצה (חוד ביצה (חוד ביצה ביצה (חוד ביצה (חוד ביצה (חוד ביצה ביצה (חוד ביצה ביצה (חוד ביצה (חוד ביצה (חוד ביצה ביצה (חוד		· ·	
וnorder (node, L) איברי הרשימה (העץ) לפי (הפעולות קורות ב- (0) . מכיוון שכל שאר הפעולות קורות ב- (0) . מכיוון שכל שאר הפעולות קורות ב- (0) . מכיוון שכל שאר הפונקציה רצה בסיבוכיות ((0) . מכיוון שכל שאר הריקה. הפונקציה ממוסר מבוצעת כמו באלגוריתם הפונקציה מכניסה (העץ) הדפסה נוסיף את הערך של הצומת למערך באולות ((0) . הפונקציה עושה זאת על הפונקציה הפונקציה עושה זאת על הפונקציה ((0) . מספר האיברים ברשימה ((0) . מספר האיברים ברשימה ((0) . מספר האיברים ברשימה ((0)) איברי הרשימה בסדר ((0)) הפונקציה מסדרת את ((0)) איברי הרשימה בסדר ((0)) איברי הרשימה ((0)) איברי הרשימה בסדר ((0)) איברי הרשימה ((0)) איברי הרשימה בסדר ((0)) איברי הרשימה ((0)) איברי ((0)) איברי הרשימה ((0)) איברי	listToArray()		הפונקציה קוראת לפונקציה
בסיבוכיות $O(n)$. מכיוון שכל שאר הפעולות קורות ב $O(n)$. מכיוון שכל שאר הפעולות קורות ב $O(n)$. מכיוון שכל שאר הפונקציה רצה בסיבוכיות $O(n)$. הפונקציה רצה בסיבוכיות $O(n)$. הרשימה ריקה. הרשימה רציקה. הפונקציה מכויסה בוצעת כמו באלגוריתם הפונקציה מכניסה בוצעת כמו באלגוריתם של איברי הרשימה (העץ) לפי סדר האינדקסים. הפונקציה הפונקציה עושה זאת על הפונקציה הפונקציה עושה זאת על איברי העץ. הפונקציה מחזירה את $O(n)$. הפונקציה מחזירה של איברי הרשימה (העץ) לפי סדר האינדקסים. הפונקציה מחזירה את $O(n)$. הפונקציה עושה זאת על הפונקציה הפונקציה הפונקציה מחזירה את $O(n)$. הפונקציה קוראת ל- הפונקציה מסדרת את (introder sort) איברי הרשימה בסדר (introder sort) לפי		l -	
הפעולות קורות ב- 0 , 0 רר האינדקסים (- 0), הפונקציה רצה בסיבוכיות 0 , 0 , מערך ריק אם הרשמה רצה בסיבוכיות 0 , 0 , הפונקציה מכניסה הבוצעת כמו באלגוריתם הדפסה נוסיף את הערך של הדפסה נוסיף את הערך של הצומת למערך 0 , ולכן הפונקציה הפונקציה עושה זאת על הפונקציה 0 , 0 , הפונקציה מחזירה את 0 (0) הפונקציה מחזירה את 0 (0) הפונקציה מסדרת את 0 (0) הפונקציה קוראת ל- 0 0 הפונקציה מסדרת את 0 0 (0)			
הפונקציה רצה בסיבוכיות (n) 0 . הרשימה ריקה. הרכנסה מבוצעת כמו באלגוריתם ההכנסה מבוצעת כמו באלגוריתם וinorder(node, L) את הערכים הדפסה נוסיף את הערך של הדפסה נוסיף את הערך של הפונקציה הפונקציה עושה זאת על הפונקציה (n) n 0. הפונקציה עושה זאת על הפונקציה (n) n 0. הפונקציה מחזירה את n 1 (n) n 1 הפונקציה מחזירה את n 2 (n) הפונקציה מחזירה את n 3 (n) הפונקציה מסדרת את n 4 (n) הפונקציה קוראת ל- n 4 הפונקציה מסדרת את n 4 (`` '	,
הרשימה ריקה. inorder(node, L) הפונקציה מכניסה inorder בשיעור (במקום לרשימה L את הערכים in-order בשיעור (במקום לרשימה (העץ) של איברי הרשימה (העץ) הדפסה נוסיף את הערך של dev סדר האינדקסים. הפונקציה עושה זאת על hin-order עיברי הרשימה. (10) הפונקציה מחזירה את length() מספר האיברים ברשימה. sort() הפונקציה מסדרת את (istToArray(self)	1		
וnorder(node, L) ההכנסה מבוצעת כמו באלגוריתם לרשימה L את הערכים וn-order של איברי הרשימה (העץ) הדפסה נוסיף את הערך של לפי סדר האינדקסים. הפונקציה על הפונקציה עושה זאת על הידי ריצה בסיבוכיות $O(n)$. הפונקציה מחזירה את length() מספר האיברים ברשימה. $O(1)$ מספר האיברים ברשימה. $O(1)$ הפונקציה מסדרת את listToArray(self) איברי הרשימה בסדר איברים ברשימה ווstToArray(self)	1		וופונקציוו וצוו בטיבוכיוונ (ח) ט.
יהרסר וואר במקום לרשימה L את הערכים של איברי הרשימה (העץ) של איברי הרשימה (העץ) הצומת למערך של לפי סדר האינדקסים. הפונקציה עושה זאת על ידי ריצה in-order על ידי ריצה in-order על איברי העץ. הפונקציה מחזירה את length() מספר האיברים ברשימה. מספר האיברים ברשימה. הפונקציה קוראת ל- הפונקציה מסדרת את listToArray(self)	ipordor(podo I.)		בבנסב מבועות כמו בעלנוכותם
הדפסה נוסיף את הערך של לפי סדר האינדקסים. הצומת למערך 0 ולכן הפונקציה הפונקציה עושה זאת על הפיבוכיות 0 . הפונקציה עושה זאת על ידי ריצה in-order על היברי העץ. הפונקציה מחזירה את 0 length() הפונקציה מחזירה את מספר האיברים ברשימה. מספר האיברים ברשימה. הפונקציה קוראת ל- הפונקציה מסדרת את listToArray(self)	moraer(noae, L)	•	-
הצומת למערך L) ולכן הפונקציה האינדקסים. הפונקציה עושה זאת על הפונקציה עושה זאת על ידי ריצה in-order על ידי ריצה ידי העץ. איברי העץ. הפונקציה מחזירה את $O(1)$ מספר האיברים ברשימה. $O(1)$ מספר האיברים ברשימה. הפונקציה קוראת ל- הפונקציה מסדרת את (istToArray(self) איברי הרשימה בסדר			• •
רצה בסיבוכיות $O(n)$. ידי ריצה in-order על ידי ריצה in-order ידי ריצה איברי העץ. איברי העץ. הפונקציה מחזירה את $O(1)$ מספר האיברים ברשימה. מספר האיברים ברשימה. מסור האיברים ברשימה איברים ברשימה. מסור הפונקציה קוראת ל- הפונקציה מסדרת את listToArray(self)			
ידי ריצה in-order על איברי העץ. איברי העץ. ס(1) הפונקציה מחזירה את (length() מספר האיברים ברשימה. תחילה הפונקציה קוראת ל- הפונקציה מסדרת את (sort() איברי הרשימה בסדר listToArray(self)		•	
איברי העץ. length() הפונקציה מחזירה את (0(1) מספר האיברים ברשימה. תחילה הפונקציה קוראת ל- הפונקציה מסדרת את (sort() שרצה איברי הרשימה בסדר listToArray(self)			רצה בסיבוכיות $U(n)$.
length() הפונקציה מחזירה את <i>O</i> (1) מספר האיברים ברשימה. מספר האיברים ברשימה. הפונקציה קוראת ל- הפונקציה מסדרת את listToArray(self) איברי הרשימה בסדר		•	
מספר האיברים ברשימה. תחילה הפונקציה קוראת ל- הפונקציה מסדרת את (sort() איברי הרשימה בסדר listToArray(self)		•	
תחילה הפונקציה קוראת ל- הפונקציה מסדרת את ()sort איברי הרשימה בסדר ()sistToArray(self)	length()	-	0(1)
listToArray(self), שרצה איברי הרשימה בסדר		מספר האיברים ברשימה.	
	sort()	הפונקציה מסדרת את	
· · · · · · · · · · · · · · · · · · ·		איברי הרשימה בסדר	(listToArray(self, שרצה,
בסיבוכיות $\mathcal{O}(n)$. עולה ומחזירה רשימה \mid		עולה ומחזירה רשימה	בסיבוכיות $O(n)$.
לאחר מכן ה [ׁ] פונקציה מפרידה את (עץ) חדשה.		(עץ) חדשה.	לאחר מכן הפונקציה מפרידה את
כל האיברים שהם None			

	T	
		מהאיברים שהם לא None
		בסיבוכיות $O(n)$.
		לאחר מכן הפונקציה קוראת ל-
		שרצה merge_sort(lst)
		בסיבוכיות $O(n \cdot \log n)$.
		אחר כך הפונקציה קוראת ל-
		buildTreeFromList(lst, first,
		O(n) שרצה בסיבוכיות last)
		ושומרת את הצומת שהתקבלה
		כשורש של AVLTreeList חדש.
		לבסוף הפונקציה מעדכנת את
		השדות של ה-AVLTreeList
		החדש. לשם כך הפונקציה קוראת
		י . max_node(node) -ל-
		אשר רצות min_node(node)
		בסיבוכיות ($\log n$. סה"כ נקבל
		כי הפונקציה (sort(self רצה
		בסיבוכיות ($O(n \cdot \log n)$.
permutation()	הפונקציה מחזירה רשימה	תחילה הפונקציה קוראת ל-
p()	עץ) חדשה המכילה את	listToArray(self)
	אותם האיברים של	בסיבוכיות $O(n)$.
	אוונם ווא בו ים סי הרשימה (העץ) הנוכחי	בס בול וול (א) ס. לאחר מכן הפונקציה קוראת
	בסדר אקראי. (self) בסדר א	לפונקציה (shuffle(lst, שרצה
	: K K 1102 (00)	רפוניון און פווארוס, או בווי בסיבוכיות $O(n)$.
		בס בול וול (א) ס. אחר כך הפונקציה קוראת ל-
		buildTreeFromList(lst, first,
		O(n) שרצה בסיבוכיות last)
		ושומרת את הצומת שהתקבלה
		כשורש של AVLTreeList חדש.
		לבסוף הפונקציה מעדכנת את
		השדות של ה-AVLTreeList
		החדש. לשם כך הפונקציה קוראת
		ל- (max_node(node) ו-
		אשר רצות min_node(node)
		רצוונ וואר וואר וואר וואר וואר וואר וואר ו
		בסיבוכיות (tog ti) טודע מובר כי הפונקציה (permutation(self
		רצה בסיבוכיות $O(n)$.
concat(lst)	הפונקציה מקבלת רשימה	ריקים self או st ריקים self או
concat(ist)	ומשרשר אותה אל סוף	בנוקו ה שבו וכו או ווספר הקים הפונקציה מחברת את העצים
	ומשו שו אוומר אל טוף הרשימה הנוכחית.	וופונקציוז מוובו זו אונ וועצים (אם יש מה לחבר) ומסיימת לרוץ
		(אם יש מה לחבר) ומטיימונ לדוץ – פעולה זו קוראת בסיבוכיות
	הפונקציה מחזירה את הערך המוחלט של הפרש	·
	ווערן וומווולט של וופרש הגבהים של עצי ה-AVL	O(1).אם האורך של אחד העצים הוא 1
	•	· ·
	שמוזגו.	הפונקציה תבצע insert לאיבר
		של העץ בעל האיבר היחיד אל
		תוך העץ השני ותסיים את הריצה
		$O(\log n)$ סיבוכיות $-$
		אחרת, הפונקציה שומרת את האיבר המקסימלי של self
	•	. COLL THE LINEAU TAINS

$(\operatorname{delete}(\operatorname{self.length}()-)$ וקוראת ל $(\operatorname{log} n)$ $O(\operatorname{log} n)$
עת, הפונקציה משווה בין הגובה של העצים (לאחר המחיקה) של העצים (לאחר המחיקה) אם הגובה של self שוה לגובה של self se שוה לגובה של self same (self, x, lst) joinSelfSame(self, x, lst) של sel הפונקציה קוראת ל- של sel הפונקציה קוראת ל- של sel הפונקציה קוראת ל- הגובה של self שול המגובה של שרצה בסיבוכיות (log n) אם joinSelfBigger(self, x, lst) של sel הפונקציה קוראת ל- הגובה של self אם מהגובה של שרצה בסיבוכיות (log n) המוכר (log n) של הנון (log n) השור לבסוף הפונקציה תעדכן את השדות של העץ (סיבוכיות (log n)) של הנון משרבו מקודם – של self הפונקציה רצה בסיבוכיות סיבוכיות (log n) השור מור של self ושמרבו מקודם – של on"כ הפונקציה רצה בסיבוכיות סיבוכיות (log n) ה"כ הפונקציה רצה בסיבוכיות סיבוכיות (log n) ה"כ הפונקציה רצה בסיבוכיות סיבוכיות self של מגובה העץ self ומחברת העץ self מנומה x העץ self ומחברת
של העצים (לאחר המחיקה) ומחברת אותם בהתאם: אם הגובה של self שווה לגובה של self הפונקציה קוראת ל- הגובה של soinSelfBigger(self, x, Ist) (olog n) אם (olog n) אם self שרצה בסיבוכיות (olog n) אם hash for points for
ומחברת אותם בהתאם: אם הגובה של self שווה לגובה אם הגובה של joinSelfSame(self, x, lst) joinSelfSame(self, x, lst) שרצה בסיבוכיות (1)0. של self הפונקציה קוראת ל- שרצה בסיבוכיות (3 (log n) ו אם הגובה של ioinSelfBigger(self, x, lst) של self הפונקציה קוראת ל- הגובה של self שרצה בסיבוכיות (3 (log n) ו אם ist joinSelfSmaller(self, x, lst) של הפונקציה תעדכן את שרצה בסיבוכיות (10 (log n) ו אם ist joinSelfSmaller(self, x, lst) של הפונקציה תעדכן את שרצה בסיבוכיות (10 (log n) ו אם ist joinSelfBigger(x, lst) של העץ (סיבוכיות (10 (log n) ו אם ist joinSelfBigger(x, lst) של הען (log n) ו אם ist joinSelfBigger(x, lst) של הען ל- של הען וומחברת את joinSelfBigger(x, lst) את העצים דרך הצומת x כנלמד
אם הגובה של self שווה לגובה של last שוה לגובה לובה קסוראת ל- joinSelfSame(self, x, Ist) joinSelfSame(self, x, Ist) (0.1) . שרצה בסיבוכיות (0.1) . אם הגובה של self אז און מהגובה של self קסוחSelfBigger(self, x, Ist) joinSelfBigger(self, x, Ist) שרצה בסיבוכיות (0.1) . אם (0.1) און מהגובה של (0.1) און מהגובה של (0.1) און מהגובה של (0.1) און
של sel הפונקציה קוראת ל- joinSelfSame(self, x, lst) au רצה בסיבוכיות (10). wa הגובה של self או מהגובה joinSelfBigger(self, x, lst) by joinSelfBigger(self, x, lst) joinSelfBigger(self, x, lst) ware בסיבוכיות (0 (log n)). אם lst הפונקציה קוראת ל- ware בסיבוכיות (10 (log n)). by deoin a found in the first of
joinSelfSame(self, x, lst) $0(1)$.
שרצה בסיבוכיות (1)0. אם הגובה של self א גדול מהגובה של self הפונקציה קוראת ל- שרצה בסיבוכיות (1)0. של log n (1)0. של self הפונקציה קוראת ל- הגובה של self קטן מהגובה של שרצה בסיבוכיות self אם (100. אם self אם foinSelfSigger(self, x, lst) ואור איז אור איז אור איז
שרצה בסיבוכיות (1)0. אם הגובה של self א גדול מהגובה של self הפונקציה קוראת ל- שרצה בסיבוכיות (1)0. של log n (1)0. של self הפונקציה קוראת ל- הגובה של self קטן מהגובה של שרצה בסיבוכיות self אם (100. אם self אם foinSelfSigger(self, x, lst) ואור איז אור איז אור איז
אם הגובה של self מהגובה של les גדול מהגובה של les הפונקציה קוראת ל- joinSelfBigger(self, x, lst) שרצה בסיבוכיות (log n). אם שרצה בסיבוכיות self אם הגובה של poinSelfBigger(self, x, lst). אם self אם קטן מהגובה של self קטון מהגובה של poinSelfSmaller(self, x, lst) joinSelfSmaller(self, x, lst) uרצה בסיבוכיות (log n). לבסוף הפונקציה תעדכן את hat העץ (oיבוכיות (log n)) השדות של העץ (oיבוכיות (log n)) שמחקנו של העץ (oיבוכיות (log n)) שמחקנו ושמרנו מקודם – belainceTree(node) שמחקנו ושמרנו מקודם – o'בוכיות (log n) הפונקציה מניחה כי גובה העץ poinSelfBigger(x, lst) הפונקציה מניחה כי גובה העץ self ומחברת את העצים דרך הצומת x כנלמד
joinSelfBigger(self, x, lst) שרצה בסיבוכיות $0(\log n)$. אם $0(\log n)$ אם $0(\log n)$ אם $0(\log n)$ אבובה של self אבובה של self אבובה של self אבובה של self אושר אושר אושר אושר אושר אושר אושר הפונקציה קוראת ל- $0(\log n)$ אביבוכיות $0(\log n)$ לבסוף הפונקציה תעדכן את שרצה בסיבוכיות $0(\log n)$ אביבוכיות $0(\log n)$ בשרות של העץ (סיבוכיות $0(\log n)$ בשרות שמחקנו ושמרנו מקודם שחקנו ושמרנו מקודם שחקנו ושמרנו מקודם חיבוכיות $0(\log n)$. הפונקציה רצה בסיבוכיות המן soinSelfBigger(x, lst) הפונקציה מניחה כי גובה העץ הפונקציה מחברת את self ומחברת את self ומחברת את העצים דרך הצומת x כנלמד
joinSelfBigger(self, x, lst) שרצה בסיבוכיות $0(\log n)$. אם $0(\log n)$ אם $0(\log n)$ אם $0(\log n)$ אבובה של self אבובה של self אבובה של self אבובה של self אושר אושר אושר אושר אושר אושר אושר הפונקציה קוראת ל- $0(\log n)$ אביבוכיות $0(\log n)$ לבסוף הפונקציה תעדכן את שרצה בסיבוכיות $0(\log n)$ אביבוכיות $0(\log n)$ בשרות של העץ (סיבוכיות $0(\log n)$ בשרות שמחקנו ושמרנו מקודם שחקנו ושמרנו מקודם שחקנו ושמרנו מקודם חיבוכיות $0(\log n)$. הפונקציה רצה בסיבוכיות המן soinSelfBigger(x, lst) הפונקציה מניחה כי גובה העץ הפונקציה מחברת את self ומחברת את self ומחברת את העצים דרך הצומת x כנלמד
שרצה בסיבוכיות $0(\log n)$. אם הגובה של self קטן מהגובה של self קטוnSelfSmaller(self, x, lst) poinSelfSmaller(self, x, lst). $0(\log n)$. שרצה בסיבוכיות $0(\log n)$ לבסוף הפונקציה תעדכן את השדות של העץ $0($ ביוכיות $0(0(0))$ מיבוכיות $0($ rebalanceTree(node) מיבוכיות $0($ rebalanceTree(node) שמחקנו ושמרנו מקודם $0($ שמחקנו ושמרנו מקודם $0($ or בסיבוכיות $0($ or $0($
הגובה של self קטן מהגובה של lst הפונקציה קוראת ל- הבונקציה קוראת ל- סימות (log n) (log n) לבסוף הפונקציה תעדכן את לבסוף הפונקציה תעדכן את השדות של העץ (סיבוכיות (0(1)) rebalanceTree (node) משר ל- ותקרא ל- שמחקנו ושמרנו מקודם – סיבוכיות (log n) (log n) סיבוכיות (ipinSelfBigger(x, lst) (log n) הפונקציה רצה בסיבוכיות הפונקציה מווחה כי גובה העץ ipinSelfBigger(x, lst) העץ ל-self דרך הצומת x את העצים דרך הצומת x כנלמד
או הפונקציה קוראת ל- $\log n$ (self, x, lst) ($\log n$) הערבה בסיבוכיות ($\log n$) לבסוף הפונקציה תעדכן את לבסוף הפונקציה תעדכן את השדות של העץ (סיבוכיות ($0(1)$) ($0(1)$) ($0(1)$) ($0(1)$) רואה בישר 1 0 (1 1 (1 2 (1 3 (1 3 (1 4 (1 4 (1 5
joinSelfSmaller(self, x, lst) $0(\log n)$ שרצה בסיבוכיות $0(\log n)$ אלבסוף הפונקציה תעדכן את השדות של העץ (סיבוכיות $0(1)$) ותקרא ל- rebalanceTree(node) כאשר rebalanceTree(node) הוא הצומת המקסימלית node הוא הצומת המקסימלית שמחקנו ושמרנו מקודם – $0(\log n)$ סיבוכיות $0(\log n)$ סיבוכיות $0(\log n)$ הפונקציה רצה בסיבוכיות הפונקציה מניחה כי גובה העץ הפונקציה מחברת את joinSelfBigger(x, lst) העץ ל-self מחברת את $0(\log n)$ את העצים דרך הצומת x כנלמד
שרצה בסיבוכיות $0(\log n)$. לבסוף הפונקציה תעדכן את השדות של העץ (סיבוכיות $0(1)$) ותקרא ל- rebalanceTree(node) כאשר rebalanceTree(node) כאשר האוא הצומת המקסימלית שמחקנו ושמרנו מקודם – $0(\log n)$. סיבוכיות $0(\log n)$. סה"כ הפונקציה רצה בסיבוכיות הפונקציה מניחה כי גובה העץ הפונקציה מחברת את joinSelfBigger(x, lst) העץ ל-self מגובה העץ ומחברת את self ומחברת את representation of the self and the self
לבסוף הפונקציה תעדכן את השדות של העץ (סיבוכיות $(0(1))$) השדות של העץ (סיבוכיות $(0(1))$) ותקרא ל- ותקרא ל- rebalanceTree(node) כאשר המקסימלית שמחקנו ושמרנו מקודם – $(0(\log n))$ סה"כ הפונקציה רצה בסיבוכיות $(0(\log n))$ הפונקציה מניחה כי גובה העץ הפונקציה מחברת את ipoinSelfBigger(x, lst) העץ ל-self מגובה העץ וlst מגובה העץ את self את העצים דרך הצומת x כנלמד
השדות של העץ (סיבוכיות ($0(1)$) ותקרא ל- ותקרא ל- rebalanceTree(node) כאשר הוא הצומת המקסימלית node הוא הצומת המקסימלית שמחקנו ושמרנו מקודם – סיבוכיות ($0(\log n)$. סה"כ הפונקציה רצה בסיבוכיות הפונקציה מחברת את joinSelfBigger(x, lst) הפונקציה העץ ל-self מגובה העץ העץ ל-self דרך הצומת x כנלמד
ותקרא ל- rebalanceTree(node) כאשר rebalanceTree(node) הוא הצומת המקסימלית node שמחקנו ושמרנו מקודם – סיבוכיות (log n). סה"כ הפונקציה רצה בסיבוכיות הפונקציה מניחה כי גובה העץ הפונקציה מחברת את joinSelfBigger(x, lst) העץ ל-self דרך הצומת x את self את העצים דרך הצומת x כנלמד
רפשר rebalanceTree(node) משר rebalanceTree (node) הוא הצומת המקסימלית node שמחקנו ושמרנו מקודם – $O(\log n)$. סיבוכיות $O(\log n)$. סיבוכיות $O(\log n)$. סיבוכיות הפונקציה רצה בסיבוכיות $O(\log n)$. הפונקציה מניחה כי גובה העץ הפונקציה מחברת את self ומחברת העץ ל-self ומחברת את self ומחברת את rebalanceTree (node) ומחברת ארבונים ארבונים דרך הצומת x כנלמד
חסומ הוא הצומת המקסימלית שמחקנו ושמרנו מקודם – שמחקנו ושמרנו מקודם – $O(\log n)$. סיבוכיות $O(\log n)$. סה"כ הפונקציה רצה בסיבוכיות $O(\log n)$. הפונקציה מוסה כי גובה העץ הפונקציה מחברת את joinSelfBigger(x, lst) את העצים דרך הצומת x כנלמד את self בומת x כנלמד
שמחקנו ושמרנו מקודם – סיבוכיות (log n). סיבוכיות סה"כ הפונקציה רצה בסיבוכיות סה"כ הפונקציה רצה בסיבוכיות (log n). $O(\log n)$. הפונקציה מובה העץ הפונקציה מחברת את self ומחברת העץ ל-self דרך הצומת x כנלמד
סיבוכיות $O(\log n)$. סיבוכיות ארד פסיבוכיות סה"כ הפונקציה רצה בסיבוכיות סה"כ הפונקציה רצה בסיבוכיות סה"כ הפונקציה רצה בסיבוכיות $O(\log n)$. הפונקציה מניחה כי גובה העץ הפונקציה מחברת את self ומחברת העץ ל-self דרך הצומת x כנלמד את את העצים דרך הצומת x כנלמד
סה"כ הפונקציה רצה בסיבוכיות .0 (log <i>n</i>). הפונקציה מניחה כי גובה העץ הפונקציה מחברת את joinSelfBigger(x, lst) את העץ ל-self ומחברת העץ ל-self דרך הצומת x את העצים דרך הצומת x כנלמד
joinSelfBigger(x, lst) הפונקציה מחברת את joinSelfBigger(x, lst) הפונקציה מחברת את self ומחברת העץ ל-self את העצים דרך הצומת x את העצים דרך הצומת x כנלמד
joinSelfBigger(x, lst) הפונקציה מחברת את self-א ומחברת העץ ל-self ומחברת העץ ל-self את אובה העץ אפונקציה מחברת את self את העץ ל-self את העצים דרך הצומת x את העצים דרך הצומת x כנלמד
self גדול מגובה העץ lst ומחברת העץ ל-self דרך הצומת x את העצים דרך הצומת x כנלמד
את העצים דרך הצומת x כנלמד
· ·
בשיעור (הפונקציה Join
$O(\log n)$ מהשיעור) – סיבוכיות מהשיעור מהשיעור מהשיעור מהשיעור מהשיעור מרוכה בער מרוכה בער מרוכה
joinSelfSmaller(x, lst) הפונקציה מחברת את הפונקציה העץ הפונקציה מחברת את
self קטן מגובה העץ lst ומחברת העץ ל-self דרך הצומת s
את העצים דרך הצומת x כנלמד
בשיעור (הפונקציה Join בשיעור
מהשיעור) – סיבוכיות $O(\log n)$
joinSelfSame(x, lst) הפונקציה מחברת את הפונקציה העץ הפונקציה מחברת את
self שווה לגובה העץ lst שרך הצומת self
ומחברת את העצים דרך הצומת
א כנלמד בשיעור (הפונקציה Join כנלמד בשיעור (
מהשיעור). נשים לב כי הפעם אין
צורך לרדת במורד אף עץ או לאזן
יותר מצומת אחת בעץ, ולכן
הפונקציה רצה בסיבוכיות $O(1)$
תחילה הפונקציה קוראת ל- הפונקציה מחזירה את search(val)
סיבוכיות האינדקס הראשון – listToArray(self)
ברשימה בו מופיע הערך $0(n)$
ולאחר מכן הפונקציה רצה Val או 1- אם לא קיים
בלולאת for על איברי הרשימה – אינדקס כזה.
O(n) סיבוכיות

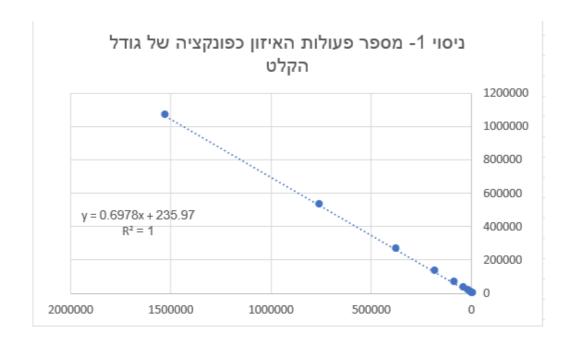
		סה"כ הפונקציה רצה בסיבוכיות $O(n)$.
rotateRight(node)	הפונקציה מבצעת סיבוב	הפונקציה פועלת כמעט כמו
3 (,	ימינה על node	שנלמד בהרצאה (לאחר
	1.000 17 1.12 12	המודיפיקציה של הוספת השדה
		<u> </u>
		height ו-height לצמתים). ההבדל
		היחיד הוא הבדיקה של האם
		node הוא השורש של העץ. אם
		הוא השורש של העץ נעדכן node
		את השורש להיות הבן השמאלי
		של $node – oיבוכיות (1)$.
		סה"כ הפונקציה רצה בסיבוכיות
		.0(1)
rotateLeft(node)	הפונקציה מבצעת סיבוב	כמעט כמו שנלמד בהרצאה
Totale Lett(Hode)	וופונזןב וו נובבעונ ט בוב שמאלה על node	לנועט כמו שנזמו בווו צאוו (לאחר המודיפיקציה של הוספת
	שנואלוו על שטטוו	
		השדה size ו-height לצמתים).
		ההבדל היחיד הוא הבדיקה של
		האם node הוא השורש של העץ.
		אם node הוא השורש של העץ
		נעדכן את השורש להיות הבן
		הימני של node – סיבוכיות
		סה"כ הפונקציה רצה $\mathcal{O}(1)$
		בסיבוכיות $0(1)$.
rebalanceTree(node)	הפונקציה מאזנת את העץ	הפונקציה זהה לחלק של האיזון
robalarios rros(nodo)	ומעדכנת את השדות כמו	של העץ של הפונקציה
	ומער כנול אול רושרוול כנור שנלמד בהרצאה	של חעץ של חופונקב וו Delete(D,x) שנלמדה בכיתה
	פנינו בווו צאוו	
		(לאחר המודיפיקציה שלכל צומת
		קיים שדה size) - סיבוכיות
		$O(\log n)$
treeSelect(i)	הפונקציות מחזירות את	הפונקציות זהות לפונקציית
treeSelectHelper(node, i)	האיבר ה-i הכי קטן בעץ	שנלמדה בכיתה Select(D, k)
	כמו שנלמד בכיתה (לאחר	לאחר המודיפיקציה שמתחילים)
	המודיפיקציה של התחלת	את החיפוש בעץ מהצומת
	החיפוש מהאיבר	$O(\log i)$ המינימלית) - סיבוכיות
	המינימלי)	
max_node(node)	הפונקציה מחזירה את	הפונקציה יורדת מהשורש של
	האיבר המקסימלי בתת-	התת עץ אל הצומת המקסימלית
	יוא בו יוניקוסינויי בונוני עץ שבו node הוא	שלו במסלול ישיר ולכן הסיבוכיות
	השורש.	0(log n) היא
min_node(node)	הפונקציה מחזירה את	הפונקציה יורדת מהשורש של
	האיבר המינימלי בתת-עץ	התת עץ אל הצומת המינימלית
	שבו node הוא השורש.	שלו במסלול ישיר ולכן הסיבוכיות
		$O(\log n)$ היא
predecessor(x)	הפונקציה מחזירה את	הפונקציה זהה לפונקציית
	האיבר הקודם ברשימה.	Predecessor(D, x) שנלמדה
	כלומר, אם x הוא האיבר	$O(\log n)$ בשיעור - סיבוכיות
	במקום ה-i הפונקציה	
	תחזיר את האיבר ה-i-1.	
	ונווויו אונווא דו וו-ו-יי	

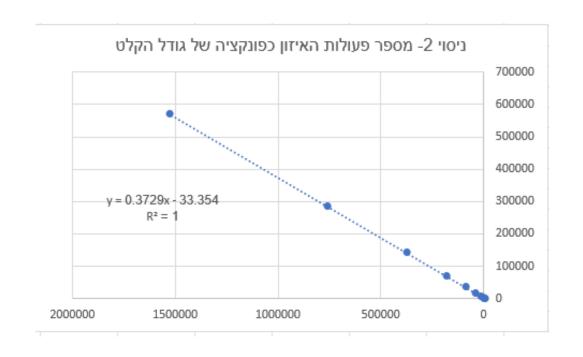
successor(x)	הפונקציה מחזירה את	הפונקציה זהה לפונקציית
	האיבר הבא ברשימה.	שנלמדה בשיעור Successor(D, x)
	כלומר, אם x הוא האיבר	$O(\log n)$ סיבוכיות -
	במקום ה-i הפונקציה	
	.i+1-תחזיר את האיבר ה	
buildTreeFromList(lst, left,	הפונקציה מקבלת רשימה	כל פעם הפונקציה מבצעת שתי
right)	של ערכים ומחזירה צומת	קריאות רקורסיביות על שני
	שמהווה שורש לעץ	חצאים שונים של הרשימה (אנו
	הנוצר AVLTreeList	עושים זאת באמצעות מצביעים
	מהרשימה.	כך שלא תיווסף סיבוכיות) ולכן
		$\log n$ עומק עץ הרקורסיה יהיה
		כאשר בכל קריאה מתבצעת
		עבודה. על כן, סיבוכיות $\mathit{O}(1)$
		הפונקציה היא
		$2^{\log n} = O(n)$
merge_sort(lst)	הפונקציות מחזירות	mergeSort הקוד זהה לקוד של
merge(lst1, lst2)	רשימה מסודרת בסדר	שלמדנו בקורס מבוא מורחב
	עולה.	$O(n \cdot \log n)$ – למדעי המחשב
Shuffle(lst)	הפונקציה "מבלגנת" את	הפונקציה עוברת על הרשימה
	in- הרשימה באופן אקראי	בלולאות for וכל פעם מחליפה בין
	.place	$\theta(n)$ - שני איברי הרשימה

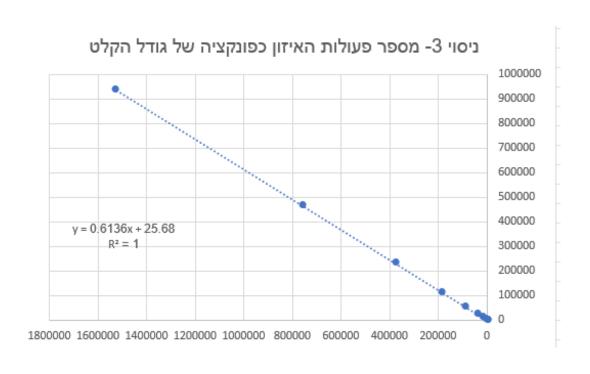
1.1. להלן טבלה המתארת את מספר פעולות האיזון שנדרשו כדי לתקן את העץ בכל ניסוי:

ניסוי 3- הכנסות ומחיקות לסירוגין	ניסוי 2- מחיקות	ניסוי 1- הכנסות	i
1830	1146	2146	1
3699	2143	4130	2
7367	4478	8332	3
14625	8933	16817	4
29600	17812	33467	5
58789	35793	67365	6
117405	71788	134828	7
236183	142998	268134	8
471642	286055	537312	9
942261	572806	1071498	10

2.1. להלן תרשימים המתארים את תוצאות הניסוי בצורה גראפית. הוספנו קו מגמה, מדד R^2 ומשוואה המתארת בקירוב את הקו. מכל אלה המסקנה היא כי הביטוי האסימפטומטי הוא (o(n) בשלושת הניסויים:







2. להלן שלושת הטבלאות המתארות את זמן הריצה בממוצע עבור עץ AVI, רשימה מקושרת ועבור מערך:

זמן ריצה- מערך הכנסות להתחלה	זמן ריצה- רשימה מקושרת הכנסות להתחלה	זמן ריצה- עץ AVL הכנסות להתחלה	i
3.295837e-07	5.654738e-07	3.664356e-05	1
5.545469e-07	6.545347e-07	2.923434e-05	2
7.467864e-07	6.878794e-07	3.192972e-05	3
9.565846e-07	6.564767e-07	3.123943e-05	4
1.113233e-06	6.747463e-07	3.225311e-05	5
1.515454e-06	6.657457e-07	3.487298e-05	6
1.724939e-06	6.754744e-07	3.123936e-05	7
1.835562e-06	6.336785e-07	3.286947e-05	8
1.964126e-06	6.112436e-07	3.112393e-05	9
2.021410e-06	6.868746e-07	3.133183e-05	10

זמן ריצה- מערך הכנסות אקראיות	זמן ריצה- רשימה מקושרת הכנסות	זמן ריצה- עץ AVL הכנסות אקראיות	i
•	אקראיות [.]	•	
8.469857e-07	1.775654e-05	4.129876e-05	1
8.965450e-07	3.748905e-05	3.700732e-05	2
1.193485e-06	7.969767e-05	3.774336e-05	3
1.327493e-06	9.021049e-04	4.012432e-05	4
1.344458e-06	1.156784e-04	4.498348e-05	5
1.230485e-06	1.331895e-04	4.744344e-05	6
1.446758e-06	1.567349e-04	4.425390e-05	7
1.546281e-06	1.936695e-04	5.345783e-05	8
1.748480e-06	2.188605e-04	4.745946e-05	9
1.645678e-06	2.324536e-04	5.449583e-05	10

זמן ריצה- מערך הכנסות בסוף	זמן ריצה- רשימה מקושרת הכנסות בסוף	זמן ריצה- עץ AVL הכנסות בסוף	i – מספר סידורי
2.241384e-07	5.812048e-07	2.802437e-05	1
1.934423e-07	6.458395e-07	2.923845e-05	2
1.748259e-07	6.835368e-07	3.001652e-05	3
1.539534e-07	6.412367e-07	3.168567e-05	4
1.934423e-07	6.503696e-07	3.432853e-05	5
1.493735e-07	6.436063e-07	3.023855e-05	6
1.495953e-07	6.130633e-07	3.359357e-05	7
1.495560e-07	6.543698e-07	3.004939e-05	8
1.535794e-07	6.663460e-07	3.583897e-05	9
1.569639e-07	6.346602e-07	3.668567e-05	10

לפני תחילת הניסוי, היינו מצפים כי התוצאות האמיתיות יהיו דומות לאלה שיצאו בסופו של דבר. ברשימה מקושרת, הכנסות לתחילתה ולסופה מתבצעות ב(1)ס (זוהי רשימה עם דבר. ברשימה מקושרת, הכנסות לתחילתה ויותר מאשר בהכנסות לסוף ולתחילת עץ AVL. מנגד, בהכנסות אקראיות ברשימה מקושרת "נאלץ" לעבור על חלק גדול מהרשימה בכל הכנסה((on) במקרה הגרוע) ולכן זמן הריצה היה איטי יותר. נשאלת השאלה מדוע במערך התקבלו התוצאות הטובות ביותר בכמעט כל המקרים? ובכן, המימוש של פייתון למערך משודרג ויעיל, ככה שקשה להתחרות בו.

הערה: במידה והיינו ממשים רשימה מקושרת ללא מצביע לסוף, אנו סבורים כי זמן הריצה היה האיטי ביותר בהכנסות לסוף, שכן במצב כזה בכל הכנסה היינו נאלצים לעבור על כל איבר ואיבר ברשימה((n) ולא במקרה הגרוע, אלא תמיד!).