מחלקת העל – WAVLTREE

<u>שדות</u>				
		private final	WAVLNode EXT	מצביע לצומת וירטואלי, אקסטרני (לשימוש פנימי בלבד). מאותחל סופית. מחוסר הורה וילדים, ובעל דרגה 1
		Private	WAVLNode root	מצביע לשורש העץ
		Private	WAVLNode max	מצביע לאיבר עם המפתח הקטן ביותר בעץ
		Private	WAVLNode min	מצביע לאיבר עם המפתח הגדול ביותר בעץ
<u>מתודות</u>				
בנאי ראשון	•	. קלט אין		
	•	יוצר עץ ריק חדש, כאשר לאחר הא	תחול השורש הוא צומת ז	וקסטרני (לשימוש עתידי).
	•	O(1) : סיבוכיות		
בנאי שני	•	קלט: מפתח מסוג int, מידע מסוג		
	•	מאתחל את השורש על ידי יצירת י		:אשר
		ס מפתח מתקבל מהקלט.		
		o מידע מתקבל מהקלט. o הורה – EXT. הצומת ו	אקסטרני (לשימוש עתידי) ואקסטרני	
	•	משנה את המצביעים min,max לי		
	•	סיבוכיות: (0(1).	וו ט (בווויוונו וויוויו י בען	
Empty	•	סיבוביווני. (1) ט. קלט: אין.		
		פלט: ערך בוליאני, האם העץ ריק	או לא	
		בכט. עון בוכיאני, וואם ווען וייקן שיטת פעולה:	.1(7 11	
			לצומת האקסטרני. EXT.	אי העץ ריק ומוחזר הערך true, על פי לוגיקת הו
		.הראשון	, , ,,	, , , , , , , , , , , , , , , , , , ,
		.false אחרת, מוחזר o		
	•	O(1) : סיבוכיות		
Search	•	קלט: מפתח מסוג int.		
	•	פלט: מחרוזת המייצגת את המידע	k של הצומת עם המפתח	
	•	שיטת פעולה:		11
			י שימוש במתודה empty. בפניע בנוצבו עד אויב ב	מוחזר null. גיעה לצומת עם מפתח הקלט, במידה ומצא – מו
		במידה ולא ריק, מבצעו את המידע ממנו.	נוויפוש בינאו יעו אשו כ	גיעוו לבוכוונ עם כופונו ו ווקלט, בכויו וו וכוצא – כוו
			ני, EXT, סימן שהמפתח	.null לא נמצא בעץ, ועל כן יוחזר
	•	.O(logn) : סיבוכיות		
Min	•	. קלט אין		
	•	פלט: מחרוזת המייצגת את המידע	של המידע של הצומת עכ	המפתח המינימלי.
	•	שיטת פעולה:		
				en, פונה לצומת בעל המפתח המינימלי בעזרת ה
		min, ומחזיר את המיד o אחרת, מחזיר null.	נ ממנו.	
	•	0(1) מיבוכיות:		
Max	•	סיבוביווני. (1) ט. קלט: אין.		
	•	פלט: מחרוזת המייצגת את המיד <i>ע</i>	. על המידע על הצומח ער	המפחח המהסימלי
	•	שיטת פעולה:		. 7/2 0 7/2/1/11/2/2/1
			, ידי שימוש במתודה pty:	en, פונה לצומת בעל המפתח המקסימלי בעזרת
		max, ומחזיר את המיז	ע ממנו.	
		o אחרת, מחזיר null.		
Ciza	•	סיבוכיות: (0(1).		
Size	•	קלט: אין.	(
	•	פלט: מספר המייצג את כמות הצנ	תים בעץ = גודל העץ.	
	•	שיטת פעולה:	מושל גידים ממו מוען למוע	יה (subTreeSize) של העצם מסוג WAVLNode
		o מחזירה את הערך המכ השורש.	פוי של גוולונונוע <mark>ל עוש</mark>	אי א LINOUE אין וועבט מיטוג (Suu I Ieesize וו
			רת היטב, מכיוון שביציו	ת הצומת האקסטרני, EXT, גודל העץ שלו מוגדר
	•	סיבוכיות: $0(1)$.		
getRoot	•	קלט: אין.		
		פלט: WAVLNode המייצג את ש		

	_	
שיטת פעולה : $ ilde{ ity}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}$	•	
o בודקת האם השורש מצביע לצומת האקסטרני, EXT, ומחזירה null אם כן. ■ יש לשים לב כי בעץ ריק השורש הוא הצומת האקסטרני.		
,		
סיבוכיות: (0(1).	•	Select
קלט: מספר של פיו נחפש את האיבר ה i בגודלו בעץ.	•	Beleet
פלט: מחרוזת המייצגת את המידע של הצומת ה i בגודלו בעץ. i	•	
שיטת פעולה:	•	
o בדיקות התחלה:		
אם העץ ריק, על ידי המתודה empty, או שאינדקס הקלט גדול מכמות הצמתים בעץ, בעזרת size המתודה size.		
יומונדרו size, מודור דעון min		
אם מונשים אוניהאבריה אשון, משמע הקטן ביהנו , מהדיי אוניהמיוע מהצומוניהמיק. בעזרת המצביע min.		
שאם מחפשים את האיבר האחרון, הבדיקה מתבצעת בעזרת המתודה size, מוחזר המידע		
מהצומת המקסימלי, בעזרת המצביע max.		
ם במידה והמתודה לא נעצרה עד כה החיפוש נעשה בשיטה הבאה:		
בכל איטרציה, נתייחס לתת העץ שאנו נמצאים בו, כאילו הוא עץ חדש, ללא זיכרון למה היה		
בעבר		
מתחילים מהשורש.		
אם גודל תת העץ השמאלי, משמע כמות המפתחות שקטנים ממני, ועוד אחד (אני) הוא		
האינדקס אותו חיפשנו, אזי שאני הצומת ה- i בגודלו בעץ.		
אחרת, עלינו להמשיך לחפש על פי:		
• אם האינדקס גדול או שווה מגודל תת העץ השמאלי, אזי סימן שהוא נמצא בתת		
העץ השמאלי, ונחזור על הפעולה עד אשר נמצא אותו.		
• אחרת, זה סימן שהוא נמצא בתת העץ הימני, אך הפעם, מכיוון ואנו מסתכלים		
על תת העץ הזה ללא זיכרון לעבר, עלינו למצוא כעת את האיבר ה-i פחות גודל		
העץ ועוד אחד, מכיוון ששללנו את יתר האיברים כבר, וכעת נחזור על הפעולה עד אשר נמצא אותו.		
אשר נמצא אודנו. \dot{i} . \dot{i} לבסוף, נחזיר את המידע של הצומת עם המפתח בגודל ה- \dot{i} .		
_ 12'		
O(logn). מיבוכיות:	•	Successor
קלט: עצם מסוג WAVLNode.	•	Successor
פלט: WAVLNode המייצג את הצומת עם המפתח הבא בגודלו בעץ, אחרי צומת הקלט.	•	
שיטת פעולה:	•	
 במידה והעצם המתקבל הוא המקסימלי בעץ, על ידי שימוש במצביע, אזי אין מפתח גדול ממנו, ויוחזר null 		
חחת. ס אחרת, למדנו כי על מנת למצוא את המפתח הבא בגודלו עלינו לרדת לבן הימני, ואז שמאלה ברציפות		
ככל האפשר, והאחרון הוא היורש.		
או מנגד, במידה ואין בן ימני, כל עוד זה בן הימני של אביו, עלינו להמשיך לטפס עד אשר נמצא אב שאני		
אהיה הבן השמאלי שלו, והאחרון יהיה היורש.		
ס לבסוף, מחזירה את היורש שמצאנו.		
O(logn) . סיבוכיות	•	
.WAVLNode קלט: עצם מסוג	•	Predecessor
פלט: WAVLNode המייצג את הצומת עם המפתח הקודם בגודלו בעץ, לפני צומת הקלט.	•	גרסא רגילה
שיטת פעולה:	•	
.false בגסראתי הבולאנית, עם החסde בגסראתי בגסראתי בגסראתי בגסראתי מפריאה לפונקציה predecessor בגסראתי ס		
.O(logn) : סיבוכיות	•	
או לא. deleteBinary ערך בוליאני המייצג האם קראנו למתודה מתוך, עצם מסוג WAVLNode, או לא.	•	Predecessor
י, עדי את הצומת עם המפתח הקודם בגודלו בעץ, לפני צומת הקלט. WAVLNode	•	גרסא בולאנית
שיטת פעולה:	•	
ים פור בכן פור. ס במידה והעצם המתקבל הוא המינימלי בעץ, על ידי שימוש במצביע, אזי אין מפתח קטן ממנו, ויוחזר		
.null		
ס אחרת, למדנו כי על מנת למצוא את המפתח הקודם בגודלו עלינו לרדת לבן השמאלי, ואז ימינה 🔾		
ברציפות ככל האפשר, והאחרון הוא הקודם בגודלו. לאורך חיפוש זה, מתבצעת בדיקה האם המתודה		
נקראה מתוך deleteBinary. במידה וכן, עלינו להוריד את גודל תת העץ עבור צומת, מכיוון שאנו עתידים למחוק את הקודם בגודלו לאחר ההחלפה.		
עונידים לבווון קאונ הקודם בגדו לו לאחור הווחלפות. או מנגד, במידה ואין בן שמאלי, כל עוד זה בן השמאלי של אביו, עלינו להמשיך לטפס עד אשר נמצא אב		
שמגיעים אליו מצד שמאל, והוא יהיה הקודם.		
ס לבסוף, מחזירה את המפתח של הקודם שמצאנו.		
O(logn) : סיבוכיות	•	
קלט: אין.	•	<u>keysToArray</u>
פלט: מערך מספרים המכיל את כל המפתחות הקיימים בעץ.	•	
שיטת פעולה:	•	

o מקבלת את הערך של גודל העץ מהפונקציה (size.		
ס במידה וגודל העץ הוא 0, מוחזר מערך ריק.		
o אחרת, נוצר מערך חדש כגודל העץ שקיבלנו קודם.		
ס כעת, מתבצעת לולאת for באורך גודל העץ של פעולות successor החל מהצומת המינימלי אשר מתקבל o		
מהמצביע, ומכניסה למערך את המפתח של כל צומת.		
ס לבסוף מוחזר המערך המלא.		
O(n). סיבוכיות	•	
טיבוניוונ: (זו)ט.	_	* . C . T . A
קלט: אין.	•	<u>infoToArray</u>
פלט: מערך מחרוזות המכיל את כל המידע הקיים בעץ.		
· · · · · · · · · · · · · · · · · · ·		
שיטת פעולה:	•	
o מקבלת את הערך של גודל העץ מהפונקציה (size).		
ס במידה וגודל העץ הוא 0, מוחזר מערך ריק.	ļ	
ס כעת, מתבצעת לולאת for באורך גודל העץ של פעולות successor החל מהצומת המינימלי אשר מתקבל o		
מהמצביע, ומכניסה למערך את המידע של כל צומת.	ļ	
ס לבסוף מוחזר המערך המלא.		
, , ,	_	
O(n) : סיבוכיות	•	
קלט: WAVLNode שהחל ממנו תתחיל הבדיקה לצורך באיזון.	•	<u>Rebalance</u>
פלט: מספר המייצג את כמות פעולות האיזון שנעשו.	•	
שיטת פעולה :	•	
המתודה תבדוק עבור כל צומת אב החל מהקלט ועד לשורש את הצורך באיזון לפי הפרמטר המוחזר 🔾		
: deleteCases מהמתודה		
אם מוחזר 0: אין צורך לבצע אף פעולת איזון וממשיכים.		
■ אם מוחזר 1: מבצעים פעולת demotion לצומת וממשיכים.		
אחרת, אנו צריכים לעקוב אחרי איזה בן לא מהווה בעיה מבחינת הפרשי דרגות, ועל כן		
אוו דו, אמי בדיכים פעקוב אוון יי איארבן פא מווחרו בפירו מברימוניום שי דו גודני ועל כן נשמור אותו בצד, בנוסף אליו, נשמור במשתנה sonDir מאיזה צד הוא הבן, ובמשתנה		
נשמור אוונו בצו, בנוטף אליו, נשמור במשוננה ווכנונס מאיזה צר הוא הבן, ובמשוננה direction את הכיוון ההפוך (לטובת שימוש עתידי ברוטציות).		
■ אם מוחזר 2: מבצעים פעולת double demotion לצומת ולבן שלו ששמרנו בשלב הקודם,		
וממשיכים.		
על singleRotation אם מוחזר 3: מדובר בפעולת רוטציה אחת, ועל כן משתמשים במתודה		
אם מורור כי מודבר בפוקני דסביר אורג, ועל כן מסונמסים במונדיד singlexolution על הבן ששמרנו עם הכיוון ההפוך מהצד בו הוא נמצא, וממשיכים.		
, , , , , , , , , , , , , , , , , , , ,		
שם מוחזר 4: מדובר בפעולת רוטציה כפולה, נפעיל את המתודה singleRotation ■		
פי הבן ששמרנו :		
• אם הוא ימני – נפעיל על בנו השמאלי של הבן, קודם רוטציה ימינה ואז שמאלה,		
וממשיכים.		
• אחרת, נפעיל על בנו הימני של הבן, קודם רוטציה שמאלה ואז ימינה, וממשיכים.		
 לבסוף, מחזירים את כמות פעולות האיזון שנעשו. 		
O(logn).	•	
(0)		singleDetation
קלט: WAVLNode המייצג את הצומת שיהפוך להיות השורש של תת העץ לאחר הרוטציה, מחרוזת המייצגת את	•	<u>singleRotation</u>
הכיוון אליו נעשה את הרוטציה.		
פלט: WAVLNode המייצג את הצומת שהיה השורש של תת העץ לפני הרוטציה.		
,		
שיטת פעולה:	•	
בתחילה, אנו שומרים 3 משתנים: הצומת שקיבלנו, אביו וגודל תת העץ שהוא שורשו.		
כעת, אנו רוצים לחבר בין האבא של השורש המקורי, לשורש החדש. נעשית בדיקה האם המקורי הוא o		
שורש העץ הגדול, ואם כן אזי שאין לו אבא, אז צריך לשנות את מצביע השורש של המחלקה להצביע,		
שוו ש הוגף הגדול, הזם כן אור שאין לו אבא, או בו ין לשנות את מבביע וושוו ש של המוזלקות להבביע, אחרת נעשית בדיקה האם ההחלפה נעשית מימין או משמאל וההשמה בהתאם.		
, , ,		
כעת, על סמָד הכיוון שקיבלנו בקלט, מתבצעת סדרת פעולות של חיבור השורש הישן כבן של השורש \circ		
החדש, ושל הבן הקודם של השורש החדש, לאביו החדש, שהוא השורש המקורי.		
כעת, מתבצעת בדיקה האם השורש המקורי הוא כעת עלה. אם כן, הוא מקבל ערכי גודל תת עץ ודרגה 🔾		
של עלה. אחרת, מוריד את דרגתו ב-1.		
o לבסוף, מחזיר את השורש המקורי.		
O(1).	•	
קלט: int המייצג את המפתח ומחרוזת המייצג את המידע אשר רוצים להכניס לעץ יחד בתור איבר חדש.	•	Insert
,	-	
פלט: מספר פעולות האיזון שהתרחשו בזמן הכנסת האיבר החדש לעץ.	•	
שיטת פעולה :	•	
חדש - 0 אם כן, מייצרים WAVLNode חדש – empty (האפשרות שהעץ ריק בעזרת המתודה (wavlnode – אם כן, מייצרים 0 אם כן		
עם הערכים שנקלטו, ומשימים אותו בתור שדה השורש של העץ, כמו גם האיבר המקסימאלי		
ומינימאלי. מחזירים 0ת שכן לא בוצעו פעולות איזון.	l	
במידה והעץ לא היה ריק, אנו עושים חיפוש בינארי על העץ, על מנת למצוא את המיקום בו צריך 🔾		
, ,		
 במידה והעץ לא היה ריק, אנו עושים חיפוש בינארי על העץ, על מנת למצוא את המיקום בו צריך להכניס את האיבר החדש (אנו יודעים זאת על ידי הגעה לעלה אקסטרני במקום הנדרש), ולשם מחברים 		
במידה והעץ לא היה ריק, אנו עושים חיפוש בינארי על העץ, על מנת למצוא את המיקום בו צריך 🔾		

ס במידה ואכן מצאנו כי ניתן להכניס את האיבר, עוברים בלולאה על כל הוריו של המקום הרצוי,		
ומגדילים להם את גודל הsubTreeSize.		
o במידה וההורה היה אונארי, אין צורך בפעולות איזון, ומוחזר 0.		
o נשמר משתנה countBalance שיספור את פעולות האיזון שמתבצעות. במידה וההורה היה עלה, נגדיל		
את הrank של ההורה ב-1, ונבדוק את שלושת מצבי האיזון עבור ההורה של ההורה (parent) ועבור כל האיברים ממנו ועד השורש (או עד שמגיעים לרוטציה) :		
■ מקרה 1: אם מגלים כי ההפרש הדרגות בין parent לבין בנו האחר הינו 1, ניתן להעלות בדרגה את parent, ולהעלות בבדיקה לאיבר הבא.		
, , , , ,		
■ מקרה 2: אם הפרשו של parent מבנו האחר הינו 2, נבדוק את הפרש הדרגות של בנו המקורי (זה שהעלנו במקור את דרגתו) מהנכד של parent אשר פונה ייפנימהיי - לכיוון בו נמצא		
יווי סוועכנו בבוקור אוג יריגומן בוונבו סע patent אסר בו בבודר (אשר כוללת גם 2). parent		
שינויי דרגה), ומחזירים את סד פעולות האיזוו.		
מקרה 3 : אם ההפרש של הבן המקורי מהנכד שפונה ייפנימהיי הינו 1, יש צורך ברוטציה		
כפולה (שכוללת 4 שינויי דרגה) ומחזירים את סך פעולות האיזון.		
כל עוד לא מגיעים לפעולת רוטציה, או לשורש, ממשיכים בבדיקה זו עבור ההורה של ההורה		
הנוכחי.		
ס לבסוף – אם לא בוצעו כלל פעולות רוטציה, מחזירים את המשתנה שספר את מספר פעולות האיזון 💿		
(אשר במקרה זה הינן רק פעולות העלאה בדרגה).		
O(logn) : סיבוכיות	•	
– קלט : מספר המייצג את המפתח של הצומת אותו עלינו למחוק.	•	<u>Delete</u>
קיסי בסבר אובר בא המרובר בא המיוון שביצענו. פלט: מספר המייצג את כמות פעולות האיזון שביצענו.	•	
בכם: בטפר רובייבג אונ כבורו פען לודר ואיאן שבי בענו. שיטת פעולה:	_	
שיטת פעולה : ס תחילה, נבדק האם העץ ריק בשימוש המתודה empty, או האם המפתח לא קיים בעץ בשימוש המתודה o	•	
י ונוילוז, נבוק האם הולך דק בשימוש המונדרה empty, או האם המפונדרלא קיים בעך בשימוש המונדרה search. אם אחד התנאים מתקיים אזי נחזיר 1		
scaten. אם אחד הונטאים פונקאים אוא נוואר ד. · • לאחר מכן, מתבצעת בדיקה האם הצומת היחידי בעץ הוא השורש, והוא גם המפתח הנדרש למחיקה,		
י לאוור מכן, מונבצעו בדיקוד ואם חבומו וידודי בעץ דווא דושון ש, ודווא גם המפונד הונדים למודיקוד, אם כן אנו מוחקים אותו על ידי הצבעה לצומת האקסטרני כעל השורש, ומחזירים כי ביצענו 0 פעולות		
אם בן אנו בוודק ביאות פל לי הבבקה לבובויניה אקסטוני בלליה היה לי ביב בלנו סבקילה ביאות. איזון.		
ס עת, אם לא נעצרנו עדיין, אנו מבצעים חיפוש בינארי עד אשר נמצא את הצומת הדרוש למחיקה, על פי o		
המפתח.		
ס כאשר מצאנו את הצומת, אנו בודקים אם הוא הצומת המינימלי או המקסימלי בעץ. במידה וכן, אנו 🔾		
מעדכנים את המצביע המתאים לצומת הבא/הקודם.		
בעת, על פי סוג הצומת עלה/אונארי/בינארי, הוא נשלח למתודת המחיקה המתאימה. \circ		
כ לאחר מכן, מבצעים פעולות rebalance במידת הצורך. \circ		
 לבסוף, מחזירים את כמות פעולות האיזון הכוללות שעשינו. 		
כ לבטון, בוויריו ים אווי בבוויוי בעו לוויריאון וובו ללווי טעטינו.		
O(logn) .	•	
, , , , , , , , , , , , , , , , , , , ,	•	<u>deleteLeaf</u>
O(logn) . סיבוכיות $WAVLNode$ המייצג את הצומת שעלינו למחוק.		<u>deleteLeaf</u>
סיבוכיות: $O(logn)$. קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו.		<u>deleteLeaf</u>
סיבוכיות: $O(logn)$. קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. שיטת פעולה:		<u>deleteLeaf</u>
סיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. שיטת פעולה: ○ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן		<u>deleteLeaf</u>
סיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. שיטת פעולה: • תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו.		<u>deleteLeaf</u>
סיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. שיטת פעולה: • תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו.		<u>deleteLeaf</u>
סיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. שיטת פעולה: ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה.		<u>deleteLeaf</u>
סיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. שיטת פעולה: ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ס לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק		<u>deleteLeaf</u>
סיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. שיטת פעולה: ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ס לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות rebalance ולהחזיר את כמות פעולות האיזון שנעשו.		<u>deleteLeaf</u>
סיבוכיות: (WAVLNode). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. פיטה מפיר המייצג את כמות פעולות האיזון שביצענו. ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ס לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות rebalance ולהחזיר את כמות פעולות האיזון שנעשו.		
סיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. סי תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. סי אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. סי לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות rebalance ולהחזיר את כמות פעולות האיזון שנעשו. סי אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות rebalance, ולהחזיר את כמות פעולות האיזון שנעשו . promotion, rotation, demotion.		deleteLeaf deleteUnary
סיבוכיות: WAVLNode המינצג את הצומת שעלינו למחוק. קלט: WAVLNode המינצג את הצומת שעלינו למחוק. פלט: מספר המינצג את כמות פעולות האיזון שביצענו. תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות Promation ולהחזיר את כמות פעולות האיזון שנעשו. אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion. סיבוכיות: (O(logn).	•	
סיבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. מוספר המייצג את כמות פעולות האיזון שביצענו. תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. אחרת, יעד סמק העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לשעות בדיקות nebalance ולהחזיר את כמות פעולות האיזון שנעשו. העושו בדיקות bromotion, rotation, demotion ולאחר מכן לבצע בדיקות promotion, rotation, demotion. מיבוכיות: O(logn). פלט: WAVLNode המייצג את הצומת שעלינו למחוק.	•	
סיבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. מוספר המייצג את כמות פעולות האיזון שביצענו. תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. אחרת, על סמך בדיקה של איזה צד הצומת הפולות האיזון שלעאו. לעשות בדיקות rebalance ולהחזיר את כמות פעולות האיזון שנעשו. אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion. סיבוכיות: (logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: WAVLNode המייצג את אביו של הנמחק.	•	
סיבוכיות: (WAVLNode). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. אחרת, יחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות aplalance ולהחזיר את כמות פעולות האיזון שנעשו. האחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion. סיבוכיות: (Ologn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: WAVLNode המייצג את אביו של הנמחק. שיטת פעולה:	•	
 סיבוכיות: (O(logn) קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ס לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות Pibalance ולהחזיר את כמות פעולות האיזון שנעשו. ס אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion. סיבוכיות: promotion, rotation, demotion. סיבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: WAVLNode המייצג את אביו של הנמחק. שיטת פעולה: המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. בתחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. 	•	
סיבוכיות: WAVLNode מלינו למחוק. פלט: WAVLNode ממייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. מודי תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. אחרת מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות rebalance ולהחזיר את כמות פעולות האיזון שנעשו. אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות rebalance, ולהחזיר את כמות פעולות האיזון שנעשו + 3, מכיוון שביצענו ברוטציה פעולת promotion, rotation, demotion. סיבוכיות: WAVLNode. קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: WAVLNode המייצג את אביו של הנמחק. שיטת פעולה: המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. המתודה למעשה מקפיצה את האבא של הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. נעשית בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים	•	
 סיבוכיות: (O(logn) קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ס לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות Pibalance ולהחזיר את כמות פעולות האיזון שנעשו. ס אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion. סיבוכיות: promotion, rotation, demotion. סיבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: WAVLNode המייצג את אביו של הנמחק. שיטת פעולה: המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. בתחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. 	•	
סיבוכיות: (WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ○ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן שיטת פעולה: ○ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ○ אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ○ לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות rebalance ולהחזיר את כמות פעולות האיזון שנעשו. ○ אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion שנילו ברוטציה פעולת הפולת mavill בריטות: promotion, rotation, demotion. □ סיבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. □ לט: WAVLNode המייצג את הצומת שעלינו למחוק. □ המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. ○ תחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. ○ עשית בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות השורש. ○ אם הוא לא השורש, אז על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב הצומת לילד הצומת.	•	
 סיבוכיות: (logn) סלט: PWAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ס לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק עלשות בדיקות rebalance ולהחזיר את כמות פעולות האיזון שנעשו. ס אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות spomotion, rotation, demotion שנעשו + 3, מכיוון שביצענו ברוטציה פעולת promotion, rotation, demotion. ס שבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: WAVLNode המייצג את אביו של הנמחק. ס המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. בתחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אוגארי, ועל פי זה מבצעים את יתר הפעולות. כ נשית בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות השורש. ס הוא לא השורש, אז על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב 	•	
סיבוכיות: (WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ○ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן □ מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ○ אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ○ לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק □ לעשות בדיקות Ebalance ולהחזיר את כמות פעולות האיזון שנעשו. ○ אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות rebalance, ולהחזיר את כמות פעולות האיזון שנעשו. □ סיבוכיות: (Pologin, rotation, demotion). □ סיבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. □ לט: WAVLNode המיצג את הצומת שעלינו למחוק. □ המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. □ בתחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. □ נעשית בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות השורש. □ אם הוא לא השורש, אז על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב הצומת לילד הצומת.	•	<u>deleteUnary</u>
סיבוכיות: (WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ○ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן שיטת פעולה: ○ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ○ אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. לעשות בדיקות הלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות robalance ולהחזיר את כמות פעולות האיזון שנעשו. ○ אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion שנעשו + 3, מכיוון שביצענו ברוטציה פעולת promotion, rotation, demotion. □ שיטת או או שביצענו ברוטציה של הנמחק הוא הבן של הנמחק. □ המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. □ עשית בדיקה האם האב הא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות השורש. □ עשית בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הום של הנמחק להיות השורש. □ אם הוא לא השורש, אז על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב הצומת לילד הצומת. □ לבסוף, מוחקים את הצומת, על ידי השמת אבא וילדים ושח, ומחזירים את צומת האב. □ לבסוף, מוחקים את הצומת, על ידי השמת אבא וילדים ושח, ומחזירים את צומת האב.	•	
סיבוכיות: (WaVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ○ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן שיטת פעולה: ○ מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ○ אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ○ לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק עשות בדיקות promotion, ולאחר מכן לבצע בדיקות שנעשו. ○ אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion שנעשו + 3, מכיוון שביצענו ברוטציה פעולת morial, ולהחזיר את כמות פעולות האיזון שנעשו. ○ שיטת שולה: □ מתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. ○ בתחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מצצעים את יתר הפעולות. ○ בתחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מורש, לכן צריך להשים את הבן של הנמחק להיות השורש. ○ בתחילה אום הנמחק להיות השורש. ○ שיטת בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות השורש, או על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב הצומת לילד הצומת. ○ לבסוף, מוחקים את הצומת, על ידי השמת אבא וילדים וlun, ומחזירים את צומת האב. □ סבוכיות: (0.0). □ לבסוף, מוחקים את הצומת שעלינו למחוק.	•	<u>deleteUnary</u>
סיבוכיות: (Viogn). סינוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. □ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. □ אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. □ לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות evalance ולהחזיר את כמות פעולות האיזון שנעשו. □ אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות rebalance, ולהחזיר את כמות פעולות האיזון שנעשו + 3, מכיוון שביצענו ברוטציה פעולת promotion, rotation, demotion. □ סיבוכיות: O(logn). □ סיבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. □ מתודה למעשה מקפיצה את האבא של הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. □ מתחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. □ מחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעת ההתאמה בין אב את הבן של הנמחק להיות השורש. □ אם הוא לא השורש, אז על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב הצומת הלילד הצומת. □ לבסוף, מוחקים את הצומת, על ידי השמת אבא וילדים ומוז, ומחזירים את צומת האב. □ סיבוכיות: (1)0. □ פלט: מספר המייצג את כמות פעולות האיזון שביצענו.	•	<u>deleteUnary</u>
סיבוכיות: (Viogn). סינוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. □ תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. □ אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. □ לאחר מחיקת העלה, מתקקימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות Promation ולהחזיר את כמות פעולות האיזון שנעשו. □ אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות rebalance, ולהחזיר את כמות פעולות האיזון שנעשו + 3, מכיוון שביצענו ברוטציה פעולת promotion, rotation, demotion. □ סיבוכיות: O(logn). □ סיבוכיות: WAVLNode המייצג את הצומת שעלינו למחוק. □ מתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. □ מתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. □ בתחילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. □ מחולה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. □ את הון של הנמחק להזיות השורש. □ לבסוף, מוחקים את הצומת, על ידי השמת אבא וילדים ומחזירים את צומת האב. □ לבסוף, מוחקים את הצומת שעלינו למחוק. □ לבסוף, מוחקים את הצומת שעלינו למחוק. □ פלט: מספר המייצג את כמות פעולות האיזון שביצענו. □ שיטת פעולה:	•	<u>deleteUnary</u>
שיבוכיות: (WaVLNode). קלט: WavLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. מסר המייצג את כמות פעולות האיזון שביצענו. מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר ו, מכיוון ששינינו את דרגתו. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. לעשות בדיקות belalance ביקוח האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות robalance ולהחזיר את כמות פעולות האיזון שנעשו. אוני בדיקות belalance ביקוח ולהחזיר את כמות פעולות האיזון שנעשו. העושו + 3, מכיוון שביצענו ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion. שינות: (O(logn). שיטת פעולה: המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק. המתודה למעשה מקפיצה את האבא של הנמחק להכיר את הבן של הנמחק הוא השורש, לכן צריך להשים עישית בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות חטורש. את הבן של הנמחק להיות חטורש. את הוא לא השורש, אז על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב הצומת לילד הצומת. מיבוכיות: (1) O. שלטר מספר המייצג את כמות פעולות האיזון שביצענו. שיטת פעולה: מימר מייצג את כמות פעולות האיזון שביצענו. מולסר משיטת פעולה: מולמר תחילה, אנו שומרים על מספר משתנים: מולמר תחילה, אנו שומרים על מספר משתנים:	•	<u>deleteUnary</u>
שיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ס לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות Promition, האיזון שנעשו. ס אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות MAVLNode. שנעשו + 3, מכיוון שביצענו ברוטציה פעולת gromotion, rotation, demotion שינשו + 4, מכיוון שביצענו ברוטציה פעולת MAVLNode. סיבוכיות: WAVLNode המיצג את אביו של הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. ס המתודה למעשה מקפיצה את האבא של הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. מעשת בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות השורש. מע הם הוא לא השורש, אז על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב אם הוא לא השורש, אז על פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב הצומת לילד הצומת. ס לבסוף, מוחקים את הצומת שעלינו למחוק. ס לבסוף, מוחקים את הצומת שעלינו למחוק. ס יבוכיות: (1) 0. σ תחילה, אנו שומרים על מספר משתנים: שיטת פעולה: ■ צומת ה-Precessor של מוסרב משתנים: ■ צומת ה-Precessor של מתודה עם הערך הבוליאני bredecessor. ■ תחילה, אנו שומרים על מספר משתנים: ■ צומת ה-Precessor של אידי קריאה למתודה עם הערך הבוליאני bredecessor.	•	<u>deleteUnary</u>
סיבוכיות: (O(lagn) קלט: WAVLNode קלט: שר המייצג את הצומת שעלינו למחוק. מספר המייצג את כמות פעולות האיזון שביצענו. מספר המייצג את כמות פעולות האיזון שביצענו. מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לשרות בחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק ועשות בדיקומו במהופאות ולאחר מכן לבצע בדיקות promotion, rotation, demotion ולאחר מכן לבצע בדיקות promotion, rotation, demotion ושנעשו + 3, מכיוון שביצענו ברוטציה פעולת מחוק. קלט: WAVLNode מיבוכיות: WAVLNode מיבוכיות: WAVLNode מיבוכיות: שלולה או מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. מיבורילה אנו מגלים מאיזה צד הצומת הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. מיבוריות: מחוקים את הצום האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות השורש. מיבוריות: (1) 0. מיבוריות: (1) 0. מיבוריות: (1) 0. מיבוריות: על מיפר משתנים: מולה: צומת ה-Stage משתנים: בפחילה, אנו שומרים על מספר משתנים: בפחילה הבור של מספר משתנים: בפחילה שלח הייצג את הצומת שרנים: במתחלה ה-Stage משתנים: במתחלה ה-Stage משתנים: במתחלה ה-Stage משתנים: במתחלה ה-Stage משתנים: במתח מת ה-Stage משתנים: במתח ה-Stage משתנים: במתח ה-Stage משתנים: במתח ה-Stage משתנים: במתח ה-Stage משתנים:	•	<u>deleteUnary</u>
שיבוכיות: (O(logn). קלט: WAVLNode המייצג את הצומת שעלינו למחוק. פלט: מספר המייצג את כמות פעולות האיזון שביצענו. ס תחילה, אנו בודקים האם האב של הצומת הנמחק הוא אונארי, אם כן, סימן שיהפוך להיות עלה, ועל כן מקבל את השמה של כל הערכים המתאימים לעלה, ומחזיר 1, מכיוון ששינינו את דרגתו. ס אחרת, על סמך בדיקה של איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת פעולה המחיקה. ס לאחר מחיקת העלה, מתקיימת בדיקה האם הבן השני של האבא הוא עלה, אם כן סימן שצריך רק לעשות בדיקות Promation ולאחר מכן לבצע בדיקות euldin האיזון שנעשו. ס אחרת, יהיה צורך ברוטציה, ולאחר מכן לבצע בדיקות promotion, rotation, demotion שנעשו + 3, מכיוון שביצענו ברוטציה פעולת mainty. ס יבוכיות: WAVLNode. סיבוכיות: WAVLNode המיצג את אביו של הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. ס המתודה למעשה מקפיצה את האבא של הנמחק הוא אונארי, ועל פי זה מבצעים את יתר הפעולות. מעשת בדיקה האם האב הוא עלה אקסטרני, ואם כן זה אומר שהנמחק הוא השורש, לכן צריך להשים את הבן של הנמחק להיות השורש. מעות הוא א השורש, אז ע פי איזה צד הצומת הנמחק הוא בן של אביו, מתבצעת ההתאמה בין אב הצומת לילד הצומת. מעות מעולה או מצולות האיזון שביצענו. מעות פעולה: מעות מעולה: בומת ה-AVAV מספר משתנים: מעות פעולה: בומת למודה על מספר משתנים: מער מומת פעולה: בומת למודה על מספר משתנים: מער מות פעולה: מער מות מומרם שלינו למתודה עם הערך הבוליאני edu.	•	<u>deleteUnary</u>

		 לבסוף, מבצעים החלפת ערכים של הצומת שעלינו למחוק, על ידי החלפת המפתח והמידע שלו בזה של 		
		ה-predecessor שלו, ובכך למעשה מבצעים את מחיקתו.		
		 לבסוף מחזירים את כמות פעולות האיזון שעשינו. 		
	•	.0(logn) : סיבוכיות		
<u>deleteCases</u>	•	קלט: WAVLNode המייצג את הצומת עליו אנו מבצעים את הבדיקה איזה תצורת איזון נדרשת לאחר המחיקה שביצענו.		
		טב בקבו. פלט: מספר המייצג את סוג תצורת המחיקה שעלינו לעשות.		
	•	שיטת פעולה:		
		המתודה למעשה מחזירה איזה מבין ארבעת המקרים האפשריים של הפרת חוקי עץ WAVL לאחר מחיקה, אם בכלל יש הפרה, קיימת עבור הצומת אותו קיבלנו.		
		תחילה, נבדקת האפשרות שאין הפרה כלל של החוקים, והמתודה תחזיר 0 המייצג שאין בעיה.		
		 לאחר מכן, בהתאם לאיזה צד ההפרה הקורית מתבצעות הבדיקות הנוספות. 		
		. אם ההפרש לבן השני הוא 2, אזי ניתן לבצע פעולת demotion בלבד, לכן נחזיר 1.		
		 אם ההפרש לבן השני הוא 1, תחילה נשמור את הבן השני במשתנה ואז: 		
		. אם ההפרש מהבן לשני בניו הוא 2, אזי double demotion יהיה מספיק, ויוחזר הערך 2.		
		אם ההפרש מהבן ה״חיצוני״ הוא 1, אזי נצטרך לבצע רוטציה, ויוחזר הערך 3.		
		■ אחרת, נצטרך לבצע רוטציה כפולה, ויוחזר הערך 4.		
	•	O(1) : סיבוכיות		
I				

מחלקה מקוננת – WAVLNode

			<u>שדות</u>
מכיל את מפתח ה-Node, כלומר הערך המספרי החחייע אשר מעיד (וקובע) את מיקום האיבר בעץ, ולפיו נבצע את המתודות השונות של עץ הWAVL. ערך זה מאותחל עייי קלט המשתמש, בעת יצירת איבר חדש.	<pre>int key</pre>	private	
מכיל את הערך שהמשתמש הגדיר עבור הnode, בעת אתחולו. ערך זה אינו חחייע.	String value	Private	7
קובע את דרגתו של הpode, בהתאם לכללי עץ WAVL, כלומר השדה תמיד יהיה בין 0 לבין פעמיים גובה העץ. בעת הכנסת איבר חדש לעץ, השדה של האיבר החדש מאותחל להיות 0, שכן הוא יהיה עלה (אשר דרגתו תמיד 0). שדה זה משמש את פעולות ההכנסה ומחיקה של העץ, על מנת לקבוע את הסיטואציה, וכן השדה מעודכן על פי הסיטואציה המתאימה.	int rank	Public	
מפנה אל הnode אשר מהווה הורה של הnode הנוכחי. השדה מאותחל באמצעות עם בניית האיבר, ע"י קלט רלוונטי. נשים לב כי ההורה של שורש העץ הינו null.	WAVLNode parent	Public	
מפנה אל בנו השמאלי של node (אשר גם הוא node). בזמן בניית הnode, השדה מאותחל להיות עלה אקסטרני EXT, שכן האיבר המדובר הינו עלה.	WAVLNode left	Public	
בדומה לשדה left, אך מצביע על הבן הימני.	WAVLNode right	Public	
שומר את גודל תת העץ (מספר האיברים) הקיים מתחת לאיבר הספציפי, כולל הוא עצמו: מתוחזק בזמן פעולות ההכנסה ומחיקה של העץ.	<pre>int subTreeSize</pre>	Public	
			<u>מתודות</u>
WAVLNode איבר מסוג, st	יוג int, ערך מסוג tring	• קלט: מפתח מס	בנאי
ימאותחל node חדש. המפתח והערך מושמים בתור מפתח וער האיבר. זפניה להורה האיבר. כמו כן, הrank של האיבר מאותחל להיות 0 (בהתאם זל להיות 1 ושדות הבנים מאותחלים להיות EXT. נשים לב כי עבור עלה הה באמצעותה כנסת מפתח 1)	LNode parent לכך שהינו עלה)		
		O(1) : סיבוכיותO(1) •	
רה את ערך השדה בסיבוכיות O(1). נשים לב כי תחזוק השגות באופן חוקי WA.	קורה במהלך בי	gets	
	קלט: איןפלט: השדה הרי		
	• פנט: השדה הודי • שיטת פעולה: מ		
'	• סיבוכיות: (1)O • סיבוכיות: (1)O		
סיבוכיות: (A) קלט: אין (מתודת מופע על איבר ספציפי)			<u>isInnerNode</u>
קלט: אין (מונודוג מופע על איבו ספציפי) פלט: ערך בולאני			
ה ה-rank האיבר גדול או שווה לאפס. אם כן, סימן שמדובר באיבר פנימי, שמדובר בעלה אקסטרני, והמתודה מחזירה false.	שיטת פעולה: ה		
	O(1) : סיבוכיות		

מדידות:

מספר פעולות האיזון המקסימלי	מספר פעולות האיזון לממוצע לפעולת delete	מספר פעולות האיזון המקסימלי	מספר פעולות האיזון הממוצע לפעולת insert	מספר פעולות	מספר סידורי
delete לפעולת	delete sirissi	insert לפעולת			
9	2.0339	15	3.4042	10,000	1
10	2.02275	18	3.38685	20,000	2
10	2.02296	18	3.43103	30,000	3
11	2.020375	19	3.41205	40,000	4
10	2.03228	18	3.40918	50,000	5
11	2.025916	18	3.41686	60,000	6
11	2.03111	18	3.40737	70,000	7
12	2.033925	19	3.41678	80,000	8
12	2.031966	19	3.40765	90,000	9
12	2.04219	19	3.40261	100,00	10

ציפיות על בסיס רקע תיאורטי: •

- ללא תלות מספר קבוע 0(1) ללא תלות מספר פעולות מספר קבוע ווnsert מספר לפעולת מספר קבוע ספר בגדול העץ או בגובהו.
- מספר פעולות האיזון המקסימלי לפעולת insert או delete צפוי להיות מסדר הגודל של גובה העץ-שכידוע מספר פעולות האיזון המקסימלי לפעולת n הוא מספר האיברים בעץ. לכן, נצפה שמספר זה יהיה מסדר הגודל 2logn כאשר n הוא מספר האיברים בעץ. של (log(n), ויגדל במתינות עם גידול כמות האיברים בעץ.

:תוצאות המדידות

- ס מספר פעולות האיזון הממוצע לפעולת insert או delete בכל אחת מהמדידות היה קבוע (עם סטיות זניחות), ולא השתנה כפונקציה של מספר האיברים בעץ. אם כך, ניתן להסיק שתוצאות אלה עולות בקנה אחד עם ניתוח ה-amortized התיאורטי של מספר פעולות האיזון עבור פעולת הכנסה ומחיקה שהתבצע בהרצאה.
- מספר פעולות האיזון המקסימלי הן עבור פעולת insert בודדת והן עבור פעולת מסדר מספר מספר מחודת היה מסדר n הוא מספר האיברים בעץ. מספר זה גדל במתינות ככל שהגדלנו את כמות הגודל של logn כאשר n הוא מספר האיברים בעץ. מספר זה גדל במתינות ככל שהגדלנו את כמות האיברים בעץ. ניתן להסיק כי גם תוצאות אלו תואמות את הניתוח התיאורטי שבוצע בכיתה.