מבני נתונים - פרויקט חקר תכנותי מספר 1- עץ דרגות

באסל ערו - 208215673

שם משתמש: baselarw

גידאא חאג׳ – 212024442

שם משתמש: gaidaahaj

4.1.2023

**תיעוד הפונקציות וסיבוכיות זמן ריצה**

**Class AVLNode:**

**שדות:**

|  |  |
| --- | --- |
| שומר את הערך שמחזיקים בנוד. | value |
| שומר את הבן השמאלי של הנוד. | left |
| שומר את הבן הימני של הנוד. | right |
| שומר את האב של הנוד. | parent |
| שומר את הגובה לנוד. | height |
| שומר כמה בנים יש בתת עץ של הנוד. | size |

**פונקציות:**

|  |  |  |
| --- | --- | --- |
| **שם פונקציה** | **תיעוד** | **סיבוכיות** |
| setSize\_node | פונקציה שמשנה שדה הsize של הנוד. | O(1) |
| getSize\_node | פונקציה שמחזירה שדה הsize של הנוד. | O(1) |
| getBF | פונקציה שמחשבת את הbalance factor של הנוד. | O(1) |
| virtual\_node | פונקציה שמשנה את size ל 0 ואת הגובה ל 1- , כלומר מייצרת ״נוד לא אמיתי״. | O(1) |

**Class AVLTreeList:**

**שדות:**

|  |  |
| --- | --- |
| שומר גודל העץ (כמה בנים יש בו). | size |
| שומר את השורש של העץ. | root |
| שומר את הבן הכי שמאלי בעץ, כלומר האיבר הראשון ברשימה. | start |
| שומר את הבן הכי ימני בעץ, כלומר האיבר האחרון ברשימה. | end |

**פונקציות בסיסיות:**

|  |  |  |
| --- | --- | --- |
| **שם פונקציה** | **תיעוד** | **סיבוכיות** |
| setRoot | משנה את ערך השורה בעץ. | O(1) |
| getRoot | מחזירה את ערך השורה בעץ. | O(1) |
| getSize | מחזירה את גודל העץ. | O(1) |
| setSize | משנה את גודל העץ. | O(1) |

**שאר הפונקציות – פונקציות שמכילות לולאות או רקורסיה :**

**הפונקציות Tree\_Select ,Tree\_Select\_rec :**

**סיבוכיות : O(log(n))**

**הסבר וניתוח:**

\*דומה מאוד לselect שראינו בכיתה.

הפונקציה **Tree\_Select** מקבלת כקלט מספר טבעי k וצריכה לחזיר את האיבר ה k לפי ההופעה שלו ברשימה.

עושה זאת על ידי קריאה לפונקציה הריקורסיבית הפונקציה **Tree\_Select\_rec** שמקבלת נוד מספר טבעי k, ובמשתנה r שומרת את המספר הסידורי של הנוד (בהרצה הראשונה הנוד יהיה השורש) על ידי קריאה לפונקציה שמחזירה את גודל העץ של תת העץ השמאלי ומוסיפים 1 (כי מובטח לנו שכל האיברים שנמצאו בתת העץ השמאלי יופיעו לפניו ברשימה). ואז נשווה בין r לבין k אם הם שווים מחזירים את הנוד שלנו ומסיימים אם r יותר גדול עושים קריאה ריקורסיבית עם אותו k ועם בהן השמאלי של הנוד, אחרת עושים קריאה ריקורסיבית עם k-r והבן הימני של הנוד- כיוון שהוא הולך להופיע במקום k-r בתת העץ שהשורש שלו הוא הבן הימני .

סך הכל במקרה הגרוע,(אם אנחנו מחפשים את האיבר הראשון או האחרון, בעץ מגודל n) עוברים על גובה העץ פעם אחת וזה עולה O(log(n)).

**הפונקציות retrive:**

**סיבוכיות : O(log(n))**

**הסבר וניתוח:**

הפונקציה מקבלת כקלט i אינדיקס ברשימה והיא צריכה להחזיר את הערך שנמצא במקום ה i ברשימה, עושה זאת על ידי קרירה לפונקציה Tree\_Select שתיארנו למעלה, והיא בתורה לוקחת O(log(n)), לכן גם retrive לוקחת O(log(n)).

**הפונקציות insert:**

**סיבוכיות : O(log(n))**

**הסבר וניתוח:**

מקבל כקלט אינדיקס i וערך value וצריכה להוסיף לרשימה במקום ה-i את הערך value, ומחזירה את מספר פעולות האיזון שבוצעו, הוא יכול להיות 0,1 או 2.

מיצרים צומת חדש, ואז מכניסים אותו למקומו הנכון לפי פונקציית Tree\_Select. ואז מתקנים את העץ על ידי קריאה לכל מיני פונקציות עזר שמתקנות אותה (fix\_the\_Hights, rotateLeft, rotateRight, fix\_after\_rotation, fix\_the\_tree ,fix\_sizes) שננתח את הסיבוכיות שלהן בהמשך, שלא פוגעות בסיבוכיות הזמן הלוגרתמית.

**פונקציות עזר:**

|  |  |  |
| --- | --- | --- |
| **שם פונקציה** | **תיעוד** | **סיבוכיות** |
| **fix\_the\_Hights** | מתקן את הגבהים של הנודים שהושפעו מפעולת ההכנסה או המחיקה, מתחיל מהנוד שהכנסנו ועולה עד השורש. | O(log(n)) |
| **fix\_sizes** | מתקן את שדה הsize של הנודים שהושפעו מפעולת ההכנסה או המחיקה, מתחיל מהנוד שהכנסנו ועולה עד השורש. | O(log(n)) |
| **fix\_the\_tree** | מתחיל מהנוד שמחקנו או הוספנו ומתקן את העץ ועושה את ה rotations לפי הסיטואציה המתאימה לפי ה balance factor .  בעת הכנסה : עשה לכל יותר פעם אחת רוטציה ומסיים.  בעת מחיקה : ייתכן ויעשה יותר מרוטציה אחת, ולכן במקרה הכי גרוע תצטרך לעלות עד השורש. | בהכנסה : O(1)  במחיקה : O(log(n)) |
| **rotateRight** | עושים רוטציה ימינה כפי שלמדנו בכיתה.  סך הכל שינוי שדות. | O(1) |
| **rotateLeft** | עושים רוטציה שמאלית כפי שלמדנו בכיתה. סך הכל שינוי שדות. | O(1) |
| **fix\_after\_rotation** | מעדכנת את הגובה והsize של הנוד שהכנסנו/מחקנו אחרי פעולות הרוטציה, כלומר לפי המקום הסופי הנכון שלו. סך הכל שינוי שדות. | O(1) |
| **minNode** | מקבלים כקלט נוד nodeספציפי ואנחנו צריכים להחזיר את הנוד הראשון בתת העץ שהשורש שלו הוא node .  במקרה הכי גרוע (אם למשל נהיה בשורש ואז נרד כגובה העץ). וזה לוגרתמי במספר הנודים. | O(log(n)) |
| **maxNode** | מקבלים כקלט נוד nodeספציפי ואנחנו צריכים להחזיר את הנוד האחרון בתת העץ שהשורש שלו הוא node .  במקרה הכי גרוע (אם למשל נהיה בשורש ואז נרד כגובה העץ). וזה לוגרתמי במספר הנודים. | O(log(n)) |
|  |  |  |
|  |  |  |