**קובץ תיעוד**

|  |  |  |
| --- | --- | --- |
| שם מתודה | סיבוכיות | הסבר |
| Empty() | O(1) | לא מכיל לולאה/רקורסיה לכן קבוע |
| Retrieve() | logn | קריאה לפונקציית TreeSelectRec() והסיבוכיות נובעת משם. |
| TreeSelectRec() | logn | בכל קריאה רקורסיבית יש סיבוכיות זמן קבועה כי רק עושים מספר סופי של פעולות מתמטיות.  לכן הסיבוכיות תלויה בכמות הקריאות הרקורסיביות והן תלויות בגובה העץ, כיוון שעבור כל קלט הולכים לבן הימני/השמאלי.  במקרה הגרוע נגיע לעלה, כלומר יהיו כמות קריאות בגודל גובה העץ. כיוון שמדובר בעצי AVL גובה העץ יהיה logn. ולכן במקרה הגרוע יהיו logn קריאות רקורסיביות – כלומר הסיבוכיות הכוללת תהיה logn. |
| First() | Log(n) | כל פעם קוראים לבן השמאלי, ובסוף מחזירים ערך. פעולות אלו לוקחים O(1), אבל בגלל שהלולאה תלויה על גובה העץ, יש log(n) איטרציות. |
| Last() | Log(n) | פונקציה בדיוק כמו first אך עם מצביעים לימין במקום שמאל |
| getRoot() | O(1) |  |
| listToArray | O(n) | עברנו על כל צומת פעם אחת בלבד בעזרת פונקציה רקורסיבית דומה ל-tree\_walk ממבוא מורחב. בכל קריה רקורסיבית מתבצע פעולה של append, וזה מתקיים בזמן קבוע. אז סהכ יש n קריאות עם סיבוכיות O(1)- סהכ O(n) |
| Length() | O(1) | לא מכיל לולאה/רקורסיה לכן קבוע |
| Tree\_Rank() | O(log(n)) | בכל רמה בעץ אנחנו מבזבזים זמן קבוע לכן הסיבוכיות תלויה בגובה העץ- log(n) |
| Search() | O(n) | עוברים על כל הצמתים בעץ בפונקציה רקורסיבית פנימית ובכל קריאה רקורסיבית יש פעולות שמתקיימות בזמן קבוע. לכן יש n קריאות של O(1). לכן סיבוכיות הרקורסיה הינה O(n).  בסוף קוראים לtree\_rank אם נמצא הערך המבוקש. סיבוכיות הקריאה הינה O(log(n)). אז מטענה הסיבוכיות הכוללת זה O(n). |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**מתודות של העצי -AVL ניתוח סיבוכיות**