**סיבוכיות:**

empty: O(1)

search: O(logn)

הפונקציה findNearestNode, לוקחת O(logn) זמן, מכיוון שהיא מתחילה בראש העץ ויכולה לרדת עד הקצה, וגובה העץ הוא O(logn)

min, max: O(1)

KeysToArray: O(n)

.המתודה הרקורסיבית שנקראת ממנה תעבור על כל אחד מהקודקודים בעץ בדיוק פעם אחת

infoToArray: O(n)

delete, insert : O(logn)

הקריאה לfindNearestNode לוקחת O(logn) זמן, ולאחר מכן פעולות האיזון לוקחות logn בworst case, וO(1) בamortized.

טבלת מדידות :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר סידורי | מספר פעולות | מספר פעולות האיזון הממוצע לפעולת insert | מספר פעולות האיזון הממוצע לפעולת delete | מספר פעולות האיזון המקסימלי לפעולת delete | מספר פעולות האיזון המקסימלי לפעולת insert |
| 1 | 10000 | 3.41 | 3 | 12 | 15 |
| 2 | 20000 | 3.42 | 3 | 14 | 18 |
| 3 | 30000 | 3.40 | 3 | 14 | 17 |
| 4 | 40000 | 3.41 | 3 | 14 | 17 |
| 5 | 50000 | 3.39 | 3 | 13 | 18 |
| 6 | 60000 | 3.41 | 3 | 14 | 18 |
| 7 | 70000 | 3.41 | 3 | 14 | 18 |
| 8 | 80000 | 3.41 | 3 | 14 | 20 |
| 9 | 90000 | 3.41 | 3 | 15 | 19 |
| 10 | 100000 | 3.4 | 3 | 15 | 21 |

למדנו בכיתה שמספר הפעולות הamortized של פעולות rebalance בפעולות insert וdelete הוא

O(1)

לכן ציפינו שמספר הפעולות הממוצע לאורך הריצות ישאר פחות או יותר קבוע.

מספר הפעולות המקסימלי האפשרי הוא O(logn), מכיוון שזהו גובה העץ, ולכל רמה בעץ מתבצע מספר סופי של פעולות איזון, לכן ציפינו שמספר הפעולות המקסימלי יהיה O(logn)

ניתן לראות כי אכן מספר הפעולות הממוצע נשאר קבוע, ומספר הפעולות המקסימלי עולה בקצב איטי, שנראה כמו O(logn)