**פרויקט 2 מבני נתונים - ערימה בינומית**

username1 - almoghaviv

id1 - 207298720

name1 - Almog Haviv

username2 - tal9

id2 - 322539651

name2 - Tal Cohen

1. **חלק מעשי - ניתוח סיבוכיות הזמן**

class AVLNode

במחלקה זו, הבנאי, פונקציות get ופונקציות set רצות בזמן קבוע.

בנוסף, קיימת במחלקה מתודה נוספת: is real node - אשר בודקת האם גובה של צומת

הוא 1 -, במידה וכן מחזירה false ואחרת true , מתודה זו רצה גם כן בזמן קבוע.

class AVLTree

1. **בנאי -**

זמן הריצה : 𝑶(𝟏)

נימוק: מעדכן את שורש העץ להיות צומת וירטואלי שמפתחו וערכו None וגובהו 1 .-

בנוסף מעדכן את גודל העץ להיות 0 . אלו הן פעולות אשר לוקחות זמן קבוע.

**ב**

**. search -**

זמן הריצה: 𝑶(𝒍𝒐𝒈(𝒏))

נימוק: נתחיל מהשורש, כל עוד לא הגענו לצומת וירטואלי בעץ שמפתחו None נבדוק

האם ערך המפתח של הצומת שווה למפתח הקלט, במידה וכן אנו מחזירים את הצומת.

אחרת נבדוק אם המפתח של הצומת גדול או שווה לקלט ונתקדם במורד העץ בהתאם

לגודל המפתח, כל השוואת מפתחות בצומת מתרחשת בזמן קבוע.

במקרה הגרוע ירדנו בכל מורד העץ כלומר מספר הפעולות בדיקה הן כגובה העץ,

מכיוון וזהו עץ AVL אז נבצע במקרה הגרוע O(log(n)) איטרציות שבכל איטרציה

העבודה היא מסדר O(1) עבודה ולכן בסך הכל הסיבוכיות היא לוגריתמית.

**ג**

**. Insert\_to\_bst -**

זמן הריצה: 𝑶(𝒍𝒐𝒈(𝒏))

נימוק: מתודת עזר זו הינה רקורסיבית, תנאי העצירה - במידה והגענו לצומת

וירטואלי. בעת הגעה לצומת כזו מתבצעת קריאה לפונקציה אשר מבצעת בעבודה

קבועה בנייה של צומת בעלת שני ילדים וירטואלים ומעדכנת את גובה הצומת להיות

0 . הקריאה הרקורסיבית- בכל פעם מתבצעת קריאה אחת, או על בן ימין או על בן

שמאל בהתאם לערך המפתח. סה"כ לכל היותר יתבצעו קריאות כגובה העץ, בכל

קריאה כמות העבודה הינה קבועה )השוואת מפתחות( ולכן סה"כ הזמן הינו

O(log(n)) .

**חלק שני -ניסוי**

כל ניסוי בוצע 10 פעמים ומיצענו את התוצאות כך שנקבל את הדיוק הגבוה ביותר.

**ניסוי ראשון -** הכנסות סדורות

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **סכום דרגות הצמתים שמחקנו** | **מספר העצים בסיום** | **מספר החיבורים הכולל** | **זמן ריצה**  **(מילישניות)** | **מספר סידורי** |
| 0 | 5 | 723 | 0.3 | 1 |
| 0 | 4 | 2182 | 0.4 | 2 |
| 0 | 5 | 6555 | 0.9 | 3 |
| 0 | 7 | 19675 | 1.7 | 4 |
| 0 | 8 | 59040 | 6.6 | 5 |
| 0 | 12 | 177134 | 18.3 | 6 |

**ניסוי שני -** הכנסות רנדומיות ומחיקת מחצית מהאיברים לפי המינימום

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **סכום דרגות הצמתים שמחקנו** | **מספר העצים בסיום** | **מספר החיבורים הכולל** | **זמן ריצה**  **(מילישניות)** | **מספר סידורי** |
| 2916.9 | 5 | 3275.9 | 0.5 | 1 |
| 10409.7 | 4 | 11498.7 | 1.9 | 2 |
| 36519.7 | 5 | 39794.7 | 4 | 3 |
| 125360.3 | 7 | 135194.3 | 11.6 | 4 |
| 421319.5 | 8 | 450835.5 | 38.9 | 5 |
| 1410577.8 | 12 | 1499138.8 | 143.3 | 6 |

**ניסוי שלישי -** הכנסה בסדר יורד ומחיקה עד שנשארים*צמתים*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **סכום דרגות הצמתים שמחקנו** | **מספר העצים בסיום** | **מספר החיבורים הכולל** | **זמן ריצה**  **(מילישניות)** | **מספר סידורי** |
| 697 | 5 | 723 | 0.4 | 1 |
| 2156 | 5 | 2182 | 1.0 | 2 |
| 6529 | 5 | 6555 | 2.8 | 3 |
| 19649 | 5 | 19675 | 4.3 | 4 |
| 59014 | 5 | 59040 | 14.9 | 5 |
| 177108 | 5 | 177134 | 40.1 | 6 |

**שאלה 2**

* **ניסוי ראשון:** 
  + **ניתוח סיבוכיות -:** הכנסה של n איברים למבנה ריק, תדרוש לבצע n פעולות insert וכפי שהסברנו בניתוח הסיבוכיות של insert, הפעולה היחידה שתדרוש סיבוכיות שהיא לא קבועה היא פעולת ה-meld. נדרש לבצע פעולות meld שסך העבודה בהם יהיה זהה לכמות הפעולות הנדרשות בהגדלת מונה בינארי n פעמים ב-1. וראינו בתרגול שביצוע n פעולות של הגדלת מונה בינארי ב-1 היא לינארית, ולכן הסיבוכיות היא לינארית.
  + **משוואה:** כמות הצמתים מיוצגת ע"י n.
* ראשית, נשים לב שמספר הקשות מייצג את כמות הלינקים שיש בגרף, זאת משום שכל לינק שמבוצע מוסיף קשת לגרף, שכן אנו מחברים שני עצים בינומיים מגודל זהה בעזרת קשת אחת בלבד.
* בנוסף, בכל עץ כמות הקשתות היא ככמות הצמתים בעץ פחות אחד זה נובע מכך שלכל צומת יש קשת אחת ויחידה לאבא שלה, ולשורש אין אבא, ולכן אין קשת כלפי מעלה ומכיוון שיש רק שורש אחד אז באמת נקבל את כמות הצמתים בעץ פחות אחד.
* לכן, אם ניקח את מספר הקשתות בערימה, ונוסיף את מספר השורשים בערימה נגיע למספר הצמתים הכולל. כלומר, בהינתן k עצים אז כמות השורשים היא k בהתאמה, ומההבחנה שביצענו למעלה כמות הקשתות היא n-k שזוהי גם כמות הלינקים ולסיכום המשוואה היא:
* **ניסוי שני:** 
  + **ניתוח סיבוכיות -:** הכנסה של n כפי שניתחנו בניסוי 1 היא O(n). כעת נבצע מחיקות. בכל מחיקה, נצטרך למצוא מינימום חדש מבין כלל העצים האחרים, כמות העצים במקרה הגרוע ביותר היא מסדר לוגריתמי במספר האיברים ולכן במקרה הגרוע, מציאת המינימום בכל מחיקה היא בסיבוכיות לוגריתמית.

בנוסף, במידה ולמינימום יש ילדים, נמצא את המינימלי מבין הילדים (במקרה הגרוע גם כן לוגריתמי ב-n) ולאחר מכן לבצע פעולת Meld עם שאר העצים בערימה. פעולה זו היא גם כן מסיבוכיות לוגריתמית לכל היותר.

לסיכום, הסיבוכיות של כל מחיקה תהייה מסדר לוגריתמי ובסך הכל מכיוון שאנו מבצעים מחיקות, נקבל כי סיבוכיות הניסוי היא

* + **משוואה:** כמות הצמתים לפני המחיקות מיוצגת ע"י n. נסביר מדוע מספר הלינקים הכולל פחות סכום הדרגות שנמחקו שווה ממש למספר הקשתות בעץ לאחר המחיקות.
    - **הבחנה ראשונה**: מספר הלינקים הוא כמספר הקשתות שנוספו לעץ בסך הכל (מתבסס על הניתוח של ניסוי 1).
    - **הבחנה שנייה**: כאשר מוחקים צומת מדרגה k, מוחקים k קשתות (לפני ביצוע פעולת meld).
    - **לכן,** מספר הקשות בעץ לאחר המחיקה הוא כמספר הקשתות שהוספנו (מס' הלינקים) פחות מספר הקשתות שמחקנו (דרגות הצמתים שנמחקו).

**לסיכום,** לאחר מחיקת צמתים קיבלנו כי ולפי המשוואה מניסוי 1 נקבל כי מספר הצמתים בסיום הוא כמספר הקשתות בסיום ועוד מספר העצים:

* **ניסוי שלוש:** 
  + **ניתוח סיבוכיות -:** הכנסה של n כפי שניתחנו בניסוי 1 היא . כעת נבצע

מחיקות (שזה מסדר מחיקות). למרות, שהמינימום הוא תמיד האיבר השמאלי ביותר בעץ הקטן ביותר, הקוד עובר על כלל השורשים של הערימה על מנת למצוא את המינימום ולכן יידרש בזמן לוגריתמי על מנת למצוא את המינימום החדש, ומכאן הניתוח האסימפטוטי יהיה זהה לניסוי השני.

* + **משוואה:** הקשר זהה לניסוי השני ולכן המשוואה זהה.
  + **נימוק הדרגות שנמחקו:** סכום הדרגות הוא כמספר הקשתות שנמחקו.
    - בערימה מגודל איברים יש חמישה עצים בינומיים בערימה מדרגות 0-4. ולכן מהמשוואה נובע כי מספר הקשתות בערימה הסופית הוא
    - נבחין, כי עקב ההכנסה הדטרמיניסטית ההפוכה אנו תמיד מוחקים את הצומת השמאלית ביותר, מכיוון שזוהי הצומת שנמחקת לא נוצרים לינקים חדשים. לכן, מספר הלינקים הכולל הוא מספר הלינקים שביצענו בהכנסה. שהוא כמספר הקשתות בעץ לפני המחיקות.
    - על מנת להגיע ל- קשתותמהערימה בגודל n נצטרך למחוק קשתות עד שנגיע למספר זה.
    - אזי, כמות הקשתות שנמחק היא שזהו בדיוק סכום מספר הדרגות שנמחקו לפי העברת אגפים במשוואה.