**תרגיל מעשי 1- מבני נתונים**

**חלק תיאורטי**

**עודי אגבאריה 212609440 odaiagbaria1**

**תאמר עבד אלרזאק 212186407**

**הערה: בסוף הקובץ יש תיעוד לפונקציות שהשתמשנו בהם, כל פונקציה עם סיבוכיות זמן הריצה שלה במקרה הגרוע.**

**שאלה 1:**

**סעיף 1.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי i | n | ניסוי 1-הכנסות | ניסוי 2-מחיקות | ניסוי 3 - הכנסות ומחיקות לסירוגין |
| 1 | **3000** | **2032** | **2618** | **(1059, 641)** |
| 2 | **6000** | **4186** | **5346** | **(2096, 1253)** |
| 3 | **12000** | **8350** | **10740** | **(4214, 2688)** |
| 4 | **24000** | **16977** | **21299** | **(8460, 5196)** |
| 5 | **48000** | **33602** | **42579** | **(16736, 10412)** |
| 6 | **96000** | **67073** | **84842** | **(33635, 21267)** |
| 7 | **192000** | **134117** | **169614** | **(66841, 42481)** |
| 8 | **384000** | **267830** | **339368** | **(134599, 84357)** |
| 9 | **768000** | **536733** | **677625** | **(267897, 167920)** |
| 10 | **1536000** | **1073713** | **1360064** | **(535607, 336630)** |

**הערה**: בעמודה השלישית המספר השמאלי הינו כמות צעדי האיזון בהכנסות הראשונות, והמספר הימני הינו כמות צעדי האיזון בפעולות ההכנסה-מחיקה(לסירוגין).

**סעיף 2.**

רואים מהטבלה שמספר צעדי האיזון הינו של כמות האיברים(שזהו כמות ההכנסות/המחיקות...).

כי קיבלנו שמספר צעדי האיזון הינו כשני שליש ממספר האיברים שהכנסנו(בערך), מספר צעדי האיזון הינו כ 88% ממספר המחיקות, מספר צעדי האיזון הינו בערך 55% מכמות הפעולות בניסוי השלישי.

**שאלה 2:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי i | n | עץ AVL הכנסות להתחלה | מערך הכנסות להתחלה | רשימה מקושרת הכנסות להתחלה |
| 1 | **1500** | **0.098195791** | **0.00138998** | **0.002178907** |
| 2 | **3000** | **0.173582315** | **0.00498724** | **0.003988743** |
| 3 | **4500** | **0.238879919** | **0.010258436** | **0.00505209** |
| 4 | **6000** | **0.352222919** | **0.015463352** | **0.005985975** |
| 5 | **75000** | **0.401303768** | **0.019947529** | **0.007988453** |
| 6 | **9000** | **0.459208965** | **0.049924374** | **0.015585661** |
| 7 | **10500** | **0.633187056** | **0.040522099** | **0.010971069** |
| 8 | **12000** | **0.600328922** | **0.060092688** | **0.015954971** |
| 9 | **13500** | **0.711441278** | **0.065454245** | **0.060838699** |
| 10 | **15000** | **0.815709114** | **0.083519697** | **0.02644515** |

**אותם נתונים אבל בממוצע(כלומר כל עמודה חילקנו את הזמן שקיבלנו בn ):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי i | n | עץ AVL הכנסות להתחלה | מערך הכנסות להתחלה | רשימה מקושרת הכנסות להתחלה |
| 1 | **1500** | **6.54639E-05** | **9.26654E-07** | **1.4526E-06** |
| 2 | **3000** | **5.78608E-05** | **1.66241E-06** | **1.32958E-06** |
| 3 | **4500** | **5.30844E-05** | **2.27965E-06** | **1.12269E-06** |
| 4 | **6000** | **5.87038E-05** | **2.57723E-06** | **9.97663E-07** |
| 5 | **75000** | **5.35072E-06** | **2.65967E-07** | **1.06513E-07** |
| 6 | **9000** | **5.10232E-05** | **5.54715E-06** | **1.73174E-06** |
| 7 | **10500** | **6.03035E-05** | **3.85925E-06** | **1.04486E-06** |
| 8 | **12000** | **5.00274E-05** | **5.00772E-06** | **1.32958E-06** |
| 9 | **13500** | **5.26994E-05** | **4.84846E-06** | **4.50657E-06** |
| 10 | **15000** | **5.43806E-05** | **5.56798E-06** | **1.76301E-06** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי i | n | עץ AVL הכנסות אקראיות | מערך הכנסות אקראיות | רשימה מקושרת הכנסות אקראיות |
| 1 | **1500** | **0.127731562** | **0.002991438** | **0.114516497** |
| 2 | **3000** | **0.146048784** | **0.003991365** | **0.313429117** |
| 3 | **4500** | **0.209882975** | **0.0104177** | **0.648750067** |
| 4 | **6000** | **0.302999258** | **0.014261246** | **1.248833418** |
| 5 | **75000** | **0.421482086** | **0.016954899** | **1.961981773** |
| 6 | **9000** | **0.477904558** | **0.025421381** | **2.905698776** |
| 7 | **10500** | **0.607182503** | **0.029919386** | **3.779091835** |
| 8 | **12000** | **0.737959385** | **0.05580163** | **4.889110088** |
| 9 | **13500** | **0.846860886** | **0.046831608** | **6.20479846** |
| 10 | **15000** | **1.038539171** | **0.052871704** | **8.019139767** |

**אותם נתונים אבל בממוצע(כלומר כל עמודה חילקנו את הזמן שקיבלנו בn ):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי i | n | עץ AVL הכנסות אקראיות | מערך הכנסות אקראיות | רשימה מקושרת הכנסות אקראיות |
| 1 | **1500** | **8.51544E-05** | **1.99429E-06** | **7.63443E-05** |
| 2 | **3000** | **4.86829E-05** | **1.33046E-06** | **0.000104476** |
| 3 | **4500** | **4.66407E-05** | **2.31504E-06** | **0.000144167** |
| 4 | **6000** | **5.04999E-05** | **2.37687E-06** | **0.000208139** |
| 5 | **75000** | **5.61976E-06** | **2.26065E-07** | **2.61598E-05** |
| 6 | **9000** | **5.31005E-05** | **2.8246E-06** | **0.000322855** |
| 7 | **10500** | **5.78269E-05** | **2.84947E-06** | **0.000359914** |
| 8 | **12000** | **6.14966E-05** | **4.65014E-06** | **0.000407426** |
| 9 | **13500** | **6.27304E-05** | **3.46901E-06** | **0.000459615** |
| 10 | **15000** | **6.92359E-05** | **3.52478E-06** | **0.000534609** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי i | n | עץ AVL הכנסות בסוף | מערך הכנסות בסוף | רשימה מקושרת הכנסות בסוף |
| 1 | **1500** | **0.115295172** | **0** | **0.405313253** |
| 2 | **3000** | **0.210729599** | **0.001078367** | **1.14580965** |
| 3 | **4500** | **0.264770746** | **0.000996828** | **2.31302309** |
| 4 | **6000** | **0.431535006** | **0.001996279** | **4.231551409** |
| 5 | **75000** | **0.452501535** | **0.002082109** | **6.584240437** |
| 6 | **9000** | **0.585441828** | **0.002617598** | **9.364143372** |
| 7 | **10500** | **0.726033449** | **0.002990484** | **12.53937483** |
| 8 | **12000** | **0.871911287** | **0.002989769** | **17.62374473** |
| 9 | **13500** | **0.877333641** | **0.003647566** | **20.78154111** |
| 10 | **15000** | **1.199523211** | **0.004127264** | **27.48762846** |

**אותם נתונים אבל בממוצע(כלומר כל עמודה חילקנו את הזמן שקיבלנו בn ):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי i | n | עץ AVL הכנסות בסוף | מערך הכנסות בסוף | רשימה מקושרת הכנסות בסוף |
| 1 | **1500** | **7.68634E-05** | **0** | **0.000270209** |
| 2 | **3000** | **7.02432E-05** | **3.59456E-07** | **0.000381937** |
| 3 | **4500** | **5.88379E-05** | **2.21517E-07** | **0.000514005** |
| 4 | **6000** | **7.19225E-05** | **3.32713E-07** | **0.000705259** |
| 5 | **75000** | **6.03335E-06** | **2.77615E-08** | **8.77899E-05** |
| 6 | **9000** | **6.50491E-05** | **2.90844E-07** | **0.00104046** |
| 7 | **10500** | **6.9146E-05** | **2.84808E-07** | **0.001194226** |
| 8 | **12000** | **7.26593E-05** | **2.49147E-07** | **0.001468645** |
| 9 | **13500** | **6.49877E-05** | **2.7019E-07** | **0.001539373** |
| 10 | **15000** | **7.99682E-05** | **2.75151E-07** | **0.001832509** |

**הסבר:**

במימוש של העץ AVL ציפינו שזה יהיה זמן לוגריתמי בn בשלושת המקרים(הכנסה בהתחלה, בסוף, בצורה אקראית).

במימוש של הרשימה המקושרת ציפינו שהכנסה בהתחלה תעלה O(1) , הכנסה בסוף O(n) , בצורה אקראית גם יעלה O(n).

במימוש של המערך ציפינו שהמימוש של פייתון משתמש במערך מעגלי וגם doubling , ואז נצפה שהכנסה בהתחלה עולה באמורטייז O(1), הכנסה בסוף(באמורטייז) גם O(1) , הכנסה אקראית תעלה O(n) .

אם מסתכלים בתוצאות, רואים שהכנסה אקראית במימוש שה המערך עולה פחות(מהרה יותר) מהכנסה אקראית במימוש של העץ שזה לא תואם לציפיות שלנו, אולי כי המימוש של המערך מובנה בפייתון ואז יעיל מאוד.

שאר הדברים כמצופה, למשל:

הכנסה בתחילת הרשימה המקושרת עולה אכן פחות מהכנסה בתחילת המימוש שלש העץ, אבל ההכנסות האקראיות וגם אלה שבסוף עולות במימוש של הרשימה המקושרת יותר מהמימוש של העץ(כמצופה).

**תיעוד לפונקציות של הקוד:**

**Empty():** סיבוכיות זמן הריצה הינה O(1), פשוט בודקים אם גודל העץ(getSize() של העץ) שווה לאפס או לא.

**Retrieve(i):** סיבוכיות זמן הריצה הינה , מחפשים צומת כרגיל בעץ AVL, אבל משום שפה אין מפתחות לכן משתמשים בתכונת ה size של כל צומת בשביל לדעת ללכת לבן הימני או השמאלי בכל פעם, לכן העלות הינה לינארית בגובה העץ שזה לוגריתמי בכמות הצמתים בעץ AVL.

**maintain\_AVL(new\_node):** סיבוכיות זמן הריצה הינה , הפונקציה מקבלת צומת (שהוספנו לאחרונה, או צומת שהינו האב של צומת שנמחק) והולכת במסלול עד השורש ועושה תיקונים כלומר:

עדכון שדות הsize של הצמתים, הגבהים, מבצעת רוטציות(ע"י קריאה לפונקציה rotate).

I**nsert(i,s):** סיבוכיות זמן הריצה הינה , מכניסים צומת כרגיל בעץ במקום הנדרש שזה , ואז הולכים במסלול עד השורש ועושים תיקונים(כלומר מעדכנים את שדות הsize של הצמתים, הגבהים, עושים רוטציות במידות הצורך).

את התיקונים עשינו ע"י קריאה לפונקצייה series\_of\_actions(new\_node) שקוראת לפונקציה maintain\_AVL(self,new\_node שמתוארת לעיל שעושה את התיקונים הנדרשים(גבהים, רוטציות...) שסיבוכיות זמן הריצה שלה גם לוגריתמית.

**Delete(i):** סיבוכיות זמן הריצה הינה ,

**First():** סיבוכיות זמן הריצה הינה O(1) , כי שמרנו מצביע על האיבר הראשון(ואז בקריאה לפונקציה החזרנו את ערכו) ודאגנו לעדכן אותו בפעולות ההכנסה/ המחיקה/ שרשור בזמן O(1) כלומר בלי לפגוע בסיבוכיות של פונקציות אלה.

**Last():**סיבוכיות זמן הריצה הינה O(1) , כי שמרנו מצביע על האיבר האחרון(ואז בקריאה לפונקציה החזרנו את ערכו) ודאגנו לעדכן אותו בפעולות ההכנסה/ המחיקה/ שרשור בזמן O(1) כלומר בלי לפגוע בסיבוכיות של פונקציות אלה.

**listToArray():**

:**arrayToList(arr)**

**length():** סיבוכיות זמן הריצה הינה O(1), כי רק מחזירים את הsize של העץ(חשוב לומר שבקוד שלנו את הsize של העץ מקבלים ע"י הפעלת הפונקציה getSize() על העץ, כלומר זאת הפונקציה שמחזירה את הערך המעודכן של הsize של העץ, את השדה size של העץ לא רלוונטי)

**Permutation():**

**Sort():** סיבוכיות זמן הריצה הינה , תחילה בנינו מערך שמתאים לעץ בעזרת הפונקציה listToArray() ואז עשינו מיון למערך זה שזה עולה ואז המרנו את המערך הממוין לעץ AVL בעזרת פונקצית arrayToList(arr) וזהו העץ החדש שלנו.

**Concat():** סיבוכיות זמן הריצה הינה , השתמשנו באותו אלגוריתם שראינו בהרצאות והתרגול, התחלנו מהשורש של העץ הגבוה מבין שני העצים שרוצים "לשרשר" עד שמצאנו הצומת הראשון עם גובה פחות קטן מh(הגובה המינימלי בין שני העצים), ואז הוספנו צומת שיקשר בין שני העצים במקום זה ועשינו תיקונים כלפי מעלה במידת הצורך, לכן זה לכל היותר לינארי בגובה העץ ואז לוגריתמי בn.

**Search(val):** סיבוכיות זמן הריצה הינה O(n), כי חייבים לרוץ על צמתי העץ(in-order), ולכן במקרה הגרוע נעבור על כולם.