**ממן 16 – מבני נתונים ומבוא לאלגוריתמים – ברנדס איתי ת.ז.**

**שאלה 1:**

*נפתור את תרגיל 11.2-2 מעמוד 193 בספר הלימוד.*

*נגדיר טבלת גיבוב שבה התנגשויות נפתרות ע"י שרשור. הטבלה מכילה 9 תאים, ומוגדרת בעזרת פונקציית הגיבוב . נכניס את רשימת המפתחות* .

*ראשית כך תראה טבלת הגיבוב:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | */* |  |  |  |  |  |  |  |  |
| *1* | */* |  |  |  |  |  |  |  |  |
| *2* | */* |  |  |  |  |  |  |  |  |
| *3* | */* |  |  |  |  |  |  |  |  |
| *4* | */* |  |  |  |  |  |  |  |  |
| *5* | */* |  |  |  |  |  |  |  |  |
| *6* | */* |  |  |  |  |  |  |  |  |
| *7* | */* |  |  |  |  |  |  |  |  |
| *8* | */* |  |  |  |  |  |  |  |  |

*נכניס את המפתח 5. מתקיים , ולכן מפתח 5 ילך לתא מספר 5 בטבלת הגיבוב:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | */* |  |  |  |  |  |  |  |  |
| *1* | */* |  |  |  |  |  |  |  |  |
| *2* | */* |  |  |  |  |  |  |  |  |
| *3* | */* |  |  |  |  |  |  |  |  |
| *4* | */* |  |  |  |  |  |  |  |  |
| *5* |  | *🡺* | *5* | *🡺* | */* |  |  |  |  |
| *6* | */* |  |  |  |  |  |  |  |  |
| *7* | */* |  |  |  |  |  |  |  |  |
| *8* | */* |  |  |  |  |  |  |  |  |

*כעת נכניס את המפתח 28. מתקיים , ולכן מפתח 28 ילך לתא מספר 1 בטבלת הגיבוב:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | */* |  |  |  |  |  |  |  |  |
| *1* | */* | *🡺* | *28* | *🡺* | */* |  |  |  |  |
| *2* | */* |  |  |  |  |  |  |  |  |
| *3* | */* |  |  |  |  |  |  |  |  |
| *4* | */* |  |  |  |  |  |  |  |  |
| *5* |  | *🡺* | *5* | *🡺* | */* |  |  |  |  |
| *6* | */* |  |  |  |  |  |  |  |  |
| *7* | */* |  |  |  |  |  |  |  |  |
| *8* | */* |  |  |  |  |  |  |  |  |

*כעת נכניס את המפתח 19. מתקיים , ולכן מפתח 19 ילך לתא מספר 1 בטבלת הגיבוב. הפעם כבר קיים ערך אחר בתא זה, ולכן מתרחשת התנגשות. התנגשות זו נפתרת ע"י שרשור המפתח החדש בראש הרשימה המקושרת:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | */* |  |  |  |  |  |  |  |  |
| *1* | */* | *🡺* | *19* | *🡺* | *28* | *🡺* | */* |  |  |
| *2* | */* |  |  |  |  |  |  |  |  |
| *3* | */* |  |  |  |  |  |  |  |  |
| *4* | */* |  |  |  |  |  |  |  |  |
| *5* |  | *🡺* | *5* | *🡺* | */* |  |  |  |  |
| *6* | */* |  |  |  |  |  |  |  |  |
| *7* | */* |  |  |  |  |  |  |  |  |
| *8* | */* |  |  |  |  |  |  |  |  |

**מעתה אציין את המשך הפיתרון בפחות פירוט, שכן כבר נימקתי את הטכניקות השונות קודם לכן.**

*מתקיים .*

*נכניס את כל המפתחות לפי הסדר אל טבלת הגיבוב, ונקבל:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *0* | */* |  |  |  |  |  |  |  |  |
| *1* | */* | *🡺* | *10* | *🡺* | *19* | *🡺* | *28* | *🡺* | */* |
| *2* | */* | *🡺* | *20* | *🡺* | */* |  |  |  |  |
| *3* | */* | *🡺* | *12* | *🡺* | */* |  |  |  |  |
| *4* | */* |  |  |  |  |  |  |  |  |
| *5* |  | *🡺* | *5* | *🡺* | */* |  |  |  |  |
| *6* | */* | *🡺* | *33* | *🡺* | *15* | *🡺* | */* |  |  |
| *7* | */* |  |  |  |  |  |  |  |  |
| *8* | */* | *🡺* | *17* | *🡺* | */* |  |  |  |  |

**שאלה 2:**

*נפתור את הבעיה 11-3 מעמוד 211 בספר הלימוד.*

1. *מספר התא שנבדוק בבדיקה (probing) ה-i נתון ע"י נוסחת הנסיגה*

*קל להראות (באינדוקציה על i, למשל) ש:*

*לכן, פונקציית הגיבוב שמצאנו מתאימה לפונקציית הגיבוב הריבועית בנוסחה 11.5 בעמוד 202 בספר הלימוד עם הפרמטרים .*

1. *עלינו להראות שבמקרה הגרוע, האלגוריתם בודק כל תא בטבלה.*

*על מנת לעשות כן, עלינו להראות שעבור כל מפתח k, כל m הבדיקות (probing) הראשונות יגובבו לערכים שונים. במילים אחרות, עלינו להראות שלכל מפתח k, ולכל i,j המקיימים*

*, מתקיים . (\*)*

*נניח בשלילה שקיימים מספרים טבעיים i,j המקיימים ומפתח k, כך ש-.*

*נניח בלי הגבלת הכלליות ש . אז .*

*מתקיים:*

*ולכן,*

העובדה ש- מתחלקת בm ללא שארית שקולה לכך ש היא כפולה שלמה כלשהי של m:

כעת, נתון ש-m היא חזקה של 2, ולכן קיים כך ש:

*כעת נבחן מהי זוגיות המכפלות j-i, i+j+1 בהינתן זוגיות i,j:*

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

*ניתן לראות של2 המכפלות לעולם אין אותה זוגיות. אם אחת זוגית, השנייה בהכרח אי-זוגית, ולהפך.*

*משום ש , אז , ומשום שk מספר שלם, נקבל*

*ש מחלק את ללא שארית.*

1. ***אם המכפלה j-i זוגית****, אז i+j+1 בהכרח מכפלה אי-זוגית.*

*i+j+1 אינו מתחלק ב (בגלל הזוגיות השונה), ולכן לפי הטענה מההדרכה לשאלה בפורום הקורס, j-i מתחלק ב-, אך זה לא אפשרי משום ש*

וזוהי סתירה.

1. ***אם המכפלה j-i אי-זוגית****, אז i+j+1 בהכרח מכפלה זוגית.*

*j-i אינו מתחלק ב (בגלל הזוגיות השונה), ולכן לפי הטענה מההדרכה לשאלה בפורום הקורס, i+j+1 מתחלק ב-, אך זה לא אפשרי משום ש*

*וזוהי סתירה.*

*לכן,* ***לא*** *קיימים מספרים טבעיים i,j המקיימים ומפתח k,*

*כך ש. התנאי (\*) מתקיים, ולכן במקרה הגרוע, האלגוריתם בודק כל תא בטבלה.*

**שאלה 3:**

*ננמק את נכונות האלגוריתם:*

*האלגוריתם המוצע זהה עבור מקרים 1 ו-2 (כמתוארים בספר הלימוד בעמוד 221 בהערת איור 12.4(\*)) ושונה רק עבור מקרה 3. נראה שאלגוריתם נכון גם עבור מקרה 3.*

*נניח שקיים עץ חיפוש בינארי T וצומת z בעלת שני בנים, אותה אנו מעוניינים למחוק. נגדיר את y להיות העוקב של z.*

*האלגוריתם מחליף בין left[y] ל left[z]. נראה שהחלפה זו לא הורסת את תכונות עץ החיפוש הבינארי.*

*y הוא העוקב של z, ולכן מקיים . מתכונת עץ החיפוש הבינארי, נקבל שכל צומת x שהוא צאצא של left[z], או left[z] עצמו, מקיים , ולכן ודאי . לפיכך, הצבת left[z] במקום left[y] תמשיך לקיים את תכונות עץ החיפוש הבינארי.*

*כמו כן, כפי שהוכח בשאלה י-6 בעמוד 176 במדריך הלמידה, לעוקב y אין בן שמאלי. לכן left[y]=NIL, וכמובן שהצבת NIL בleft[z] לא פוגעת בתכונות עץ החיפוש הבינארי.*

*חשוב לציין שעד שלב זה, מספר הצמתים בעץ נשאר זהה, שכן רק החלפנו בין המצביעים ולא הסרנו או ניתקנו אף צומת מן העץ.*

*כעת לz יש רק בן אחד (ימני), שכן הצבנו NIL בבנו השמאלי, וכעת אכן ניתן לטפל בz כמו במקרה 2 ב(\*), שנכונותו הונמקה בספר הלימוד בפירוט בעמודים 221-222 שלאחר הגדרת השגרה TREE-DELETE.*

*ולכן, האלגוריתם החלופי המוצע בשאלה מוחק את הצומת z מהעץ בהצלחה, מבלי לפגום בהגדרתו כעץ חיפוש בינארי.*

*ננתח את זמן הריצה במקרה הגרוע:*

*איתור העוקב y מתבצע בזמן של , כפי שהוכח במשפט 12.2 בעמוד 218 בספר הלימוד.*

*ההחלפה בין left[y] ל-left[z] מתבצעת בזמן קבוע.*

*לאחר מכן, אנו מוחקים צומת כמו במקרה 2 (\*), ז"א, מחברים בין האב של z ישירות אל הבן הימני של z. הנ"ל קורה בזמן קבוע.*

*לסיכום, זמן הריצה של האלגוריתם החלופי המוצע בשאלה רץ בסיבוכיות של .*

*יתרונות וחסרונות האלגוריתם:*

*יתרון – קלות הבנה*

*האלגוריתם הנ"ל הינו מאוד פשוט רעיונית, וקל יותר להבין אותו מאשר את האלגוריתם המקורי מהשגרה המקורית בספר הלימוד (עמוד 221).*

*יתרון – לא מעתיקים נתונים נלווים מיותרים*

*בשגרה המקורית בספר הלימוד (עמוד 221), בשורה 16, אנו מעתיקים את הנתונים הנלווים (satellite data). יתכן שנתונים אלה כבדים, ופעולת ההעתקה תהיה כבדה ויקרה. בניגוד לכך, באלגוריתם שבשאלה שלנו איננו מעתיקים נתונים נלווים.*

*חסרון – פגיעה באיזון העץ והגדלת זמן הריצה*

*אלגוריתם המחיקה המתואר בספר (בע"מ 221 תחת השגרה TREE-DELETE) משאיר את המסלולים מהשורש לעלים כמעט באותו הגובה, ז"א, המרחק מהשורש לכל אחד מהצמתים נשאר זהה, או קטן ב-1 עבור צמתים שהיו בנים של הצומת שמחקנו.*

*בניגוד לכך, האלגוריתם בשאלה שלנו מחליף בין התת-עץ השמאלי של הצומת הנמחק לתת-העץ השמאלי של העוקב של הצומת הנמחק.*

*יהי x צאצא של הצומת הנמחק. אם לפני פעולת המחיקה, המרחק משורש העץ אל הצומת הנמחק הוא h, המרחק מהצומת הנמחק אל x הוא , והמרחק מהצומת הנמחק אל העוקב שלו הוא , אז המרחק של x מהשורש הוא , אך לאחר פעולת המחיקה המרחק יהיה .*

*משום שכל פעולות המילון מתבססות על המרחק מהשורש אל הצמתים הרלוונטיים, פעולות המילון עלולות לעבור על יותר תאים בהשוואה ללפני פעולת המחיקה.*

*לכן, אלגוריתם המחיקה המתואר בספר מגדיל את עומק העץ, ולכן פוגע ביעילותו.*

**שאלה 4:**

1. *ע"מ להבחין בין מצביעים "אמתיים" לבין חוטים ניתן לבדוק האם האב של הבן של הצומת הוא הצומת עצמו.*

*ז"א לכל צומת z, נבדוק האם . אם כן, הבן השמאלי הוא בן "אמיתי". אחרת, הבן השמאלי הוא חוט.*

*נבדוק גם האם . אם כן, הבן הימני הוא בן "אמיתי". אחרת, הבן הימני הוא חוט.*

*אם האב של הבן של צומת איננו הצומת עצמו, הרי שבהכרח עברנו למקום אחר בעץ, בעזרת תכונת החיווט של עץ מיוחד זה.*

*כמובן שהבדיקה היא בדיקת מצביעים, ולא בדיקת מפתחות, לכן אין בעיה להשתמש בדרך זו גם בעצים שבהם המפתחות אינם ייחודיים.*

1. ***נכתוב שגרה עבור פעולת ההכנסה****. חלקה מועתק ממימוש השגרה TREE-INSERT בעמוד 220 בספר הלימוד תחת הנושא "הכנסה". חלקים השונים מהמימוש המקורי יסומנו במרקר צהוב.*

WIRED-TREE-INSERT(T, z)

1 y 🡨 NIL

2 x 🡨 root[T]

3 while p[x] == y

4 do y 🡨 x

5 if key[z] < key[x]

6 then x 🡨 left[x]

7 else

8 then x 🡨 right[x]

9 p[z] 🡨 y

10 if y == NIL

11 then root[T] 🡨 z

12 elseif key[z] < key[y]

13 then right[z] 🡨 y

14 left[z] 🡨 left[y]

15 left[y] 🡨 z

16 else

17 then left[z] 🡨 y

18 right[z] 🡨 right[y]

19 right[y] 🡨 z

***הסבר האלגוריתם:***

*בשורה 3 עלינו להחליף את הבדיקה, שכן כעת לעלה אין NIL בקצוות, אלא חוטים (מלבד, אולי, המצביע השמאלי ביותר בעץ והמצביע הימני ביותר בעץ).*

*את הבדיקה נעשה בעזרת הרעיון שהצגנו בסעיף א'. בתחילת כל איטרציה, x הוא האב של y, אז אפשר לבדוק ישירות האם . הנ"ל נכון רק אם הקשר ביניהם הוא קשר "אמיתי" ולא קשר מחווט. אם הקשר הוא קשר מחווט, הלולאה תשבר, ו y יהיה הצומת שתחתיו נרצה להכניס את הצומת החדש.*

*בשורות 13-14 ו17-18 אנו מחווטים את הצומת החדש לפי כללי העץ המחווט: את מצביע הבן הימני נפנה אל העוקב של הצומת, ואת מצביע הבן השמאלי נפנה אל הקודם של הצומת.*

*נבחין שאם הצומת החדש z הוא בן שמאלי של אביו (שורות 12-15), אז אביו y הוא העוקב שלו, שכן בסריקה תוכית, לאחר z נקרא את y.*

*עבור הקודם של z נוכל להשתמש בצומת הקודם המקורי של אביו y, שזהו בנו השמאלי (המחווט) המקורי של y. זאת משום שבסריקה תוכית לפני הכנסת הצומת z, left[y] היה הצומת הקודם של y, ז"א, הצומת בעל המפתח הגדול ביותר שקטן או שווה למפתח y.*

*לכן, אם z נכנס בתור בן שמאלי של y, מתקיים , ולכן ודאי הבן השמאלי המקורי של y מתאים להיות האיבר הקודם של z.*

*באופן סימטרי, אם הצומת החדש z הוא בן ימני של אביו (שורות 16-19), אז אביו y הוא הקודם של z בסריקה תוכית, ו המקורי הוא העוקב של z בסריקה תוכית.*

***ניתוח סיבוכיות זמן הריצה:***

*כפי שהוכח במשפט 12.3, בעמוד 222 בספר הלימוד, השגרה TREE-INSERT המקורית רצה ב-. בגרסה זו שונה שורה 3 לתנאי מקביל המתאים לתנאי העץ, אך הלולאה תרוץ אותו מספר פעמים, אז הסיבוכיות נשארת כשל הגרסה המקורית.*

*בנוסף, בגרסה שלנו רצות שורות 13-14 או 17-18, שאינן קיימות בגרסה המקורית, אך כל אחת מהן רצה בזמן קבוע.*

*לכן, סיבוכיות הזמן של שגרת ההכנסה החדשה זהה לשל השגרה המקורית - .*

***כעת, נכתוב שגרות עבור פעולת המחיקה****. באלגוריתם נשתמש בשגרה TREE-DELETE מסוף עמוד 221 בספר הלימוד. בנוסף נניח שקיימות השגרות TREE-SUCCESSOR ו-TREE-PREDECENSSOR אשר מחשבות את הצומת העוקב והצומת הקודם של צומת, בהתאמה.*

WIRED-TREE-DELETE(T, z)

1 TREE-DELETE(T, z)

► If leaf

2 if (p[right[z]] != z) and (p[left[z]] != z)

► if z is a right child of its parent

3 then if right[p[z]] == z

4 then right[p[z]] 🡨 right[z]

► if z is a left child of its parent

5 else

6 then left[p[z]] 🡨 left[z]

► If z has only a right child

7 elseif (p[right[z]] == z) and (p[left[z]] != z)

8 then left[TREE-SUCCESSOR(z)] 🡨 left[z]

► If z has only a left child

9 elseif (p[right[z]] == z) and (p[left[z]] != z)

10 then right[TREE-PREDECESSOR(z)] 🡨 right[z]

***הסבר האלגוריתם:***

*ראשית, בשורה 1, נבצע מחיקת צומת לפי השגרה הרגילה מספר הלימוד.*

*בעקבות המחיקה, יתכן שנאלץ לתקן את ה"חוטים" התקולים, ז"א, יתכן שנצטרך לעדכן את שדה הright של הצומת הקודם, ו/או את שדה הleft של הצומת העוקב, שכן יתכן שכעת הם מפנים, בעזרת קישורי ה"חוטים", אל הצומת שמחקנו הרגע מן העץ.*

*את התיקון נבצע באופן שונה לפי מספר בני הצומת הנמחק.*

*הבדיקות לגבי האם בן מסוים הוא "אמיתי" הוא "חוט" נעשים בעזרת הרעיון שהוצג בסעיף א'.*

1. ***z הוא עלה (שורות 2-6)***

*נפריד מקרים לגבי האם z הוא בן ימני או שמאלי של אביו y.*

* *אם z הוא בן ימני (שורות 3-4), אז אביו y הוא הצומת הקודם שלו בסריקה תוכית. מכאן שהצומת העוקב של y הוא z עצמו. לכן, בהיעדר z, הצומת העוקב הבא המתאים הוא הצומת העוקב של z. נעדכן את שדה הright של האב שיפנה אל הצומת העוקב של z.*

*לגבי הצומת העוקב של z, אם הוא קיים, אנו יודעים שקיים לו בן שמאלי "אמיתי":*

*משום שz עלה, כמובן שלא קיים לו תת-עץ ימני. לפי הניתוח של השגרה TREE-SUCCESSOR (בעמוד 218, פסקה שניה לאחר הגדרת השגרה), "*העוקב של z הוא האב הקדמון הנמוך ביותר של z אשר בנו השמאלי הוא גם אב קדמון של z*".*

*משום שלעוקב של z קיים בן שמאלי שהוא גם אב קדמון של z, כמובן שמסיקים שקיים לz בן שמאלי (אמיתי). (\*)*

*לכן אין צורך לבצע "תיקון" שכן זהו איננו חוט.*

* *אם z הוא בן שמאלי (שורות 5-6), ניתן לבצע ניתוח סימטרי למקרה שבו z בן ימני.*

*הפעם, אביו y הוא הצומת העוקב של z, ולכן z הוא הצומת הקודם של y. בהיעדר z, נעדכן את שדה הleft של האב שיפנה אל הצומת הקודם של z.*

*לגבי הצומת הקודם של z, אם הוא קיים, אנו יודעים שקיים לו בן ימני "אמיתי" ולכן אין צורך לבצע "תיקון" שכן זהו איננו חוט. (ההסבר המלא סימטרי לחלוטין: אם z הוא עלה, כמובן שלא קיים לו תת-עץ שמאלי, ואז הקודם של z הוא האב הקדמון הנמוך ביותר של z אשר בנו הימני הוא גם אב קדמון של z. מכאן שכמובן – קיים בן ימני לקודם של z (\*\*) ).*

1. ***לz יש בן אחד (שורות 7-10)***

*נפריד למקרים לגבי האם לz יש בן ימני או שמאלי.*

* *אם בנו של z הוא ימני (שורות 7-8), אז כמובן שאין לz בן שמאלי. אם לz יש צומת קודם בסריקה תוכית, לצומת הקודם של z יש בן ימני "אמיתי" (לפי אותה הצדקה כמו (\*\*)), ולכן אין צורך לבצע "תיקון" שכן זהו איננו חוט.*

*את הצומת העוקב של z עלינו לתקן. הצומת העוקב בסריקה תוכית הוא בנו הימני של z. בהיעדר z, נעדכן את שדה הleft של הבן כך שיפנה אל הצומת הקודם של z, שהוא כמובן left[z].*

* *אם בנו של z הוא שמאלי (שורות 9-10), ניתן לבצע ניתוח סימטרי למקרה שבו z בן ימני. כמובן שלא קיים לz בן ימני.*

*אם לz יש צומת עוקב בסריקה תוכית, לצומת העוקב של z יש בן שמאלי "אמיתי" (לפי אותה הצדקה כמו (\*)), ולכן אין צורך לבצע "תיקון" שכן זהו איננו חוט.*

*את הצומת הקודם בסריקה תוכית של z עלינו לתקן. בהיעדר z, נעדכן את שדה הright של הקודם של z כך שיפנה אל הצומת העוקב של z, שהוא כמובן right[z].*

1. ***לz יש 2 בנים***

*במקרה זה אין צורך לעשות שום תיקון. בשגרה* TREE-DELETE, לאחר מחיקת הצומת z, העוקב שלו y "תופס" את מקומו, בעזרת העתקת כל הנתונים הנלווים (שורה 16 בשגרה TREE-DELETE), ולכן לא קיימות הפניות אל התא שנמחק, ואין צורך בתיקון.

***ניתוח סיבוכיות זמן הריצה:***

*כפי שהוכח במשפט 12.3, בעמוד 222 בספר הלימוד, השגרה TREE-DELETE המקורית רצה ב-. לכן שורה 1 רצה בסיבוכיות של .*

*נבחין שרק אחת מקבוצות השורות הבאות רצות בפועל:*

* *שורות 2-6, שרצות בזמן קבוע.*
* *שורות 7-8, שמבצעות חישוב צומת עוקב בעלות של כפי שהוכח במשפט 12.2 בעמוד 218 בספר הלימוד.*
* *שורות 9-10, שמבצעות חישוב צומת קודם בעלות של כפי שהוכח במשפט 12.2 בעמוד 218 בספר הלימוד.*

*לכן, במקרה הגרוע, שורות 2-10 ירוצו בזמן של .*

*לסיכום, סיבוכיות הזמן של שגרת המחיקה החדשה זהה לשל השגרה המקורית - .*

1. *היתרון של שימוש בחוטים הוא שאם לצומת אין בן שמאלי/ימני, יהיה ניתן להגיע אל הקודם/העוקב (בהתאמה) של הצומת בזמן קבוע, במקום בזמן (כפי שהוכח במשפט 12.2 בעמוד 218 בספר הלימוד).*