שאלה 1.

א.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי | מספר חילופים ממוין הפוך | עלות החיפוש ממוין הפוך | מספר חילופים אקראי | עלות חיפוש אקראי |
| 1 | 1999000 | 36884 | 997086 | 31707 |
| 2 | 7998000 | 81764 | 4003129 | 71428 |
| 3 | 31996000 | 179524 | 16079766 | 164858 |
| 4 | 127992000 | 391044 | 64285903 | 354201 |
| 5 | 511984000 | 846084 | 256763914 | 782987 |

ב. קומבינטורית, נוכיח באינדוקציה כי מספר החילופים הוא

בסיס n=2 הסדר הוא 2,1 ולכן יש חילוף אחד,

צעד, יש את הרשימה n+1,n,n-1 …. 3, 2,1 ישנם 1+2+3….+n-1 חילופים בn האיברים האחרונים מהנחת האינדוקציה, ומכיוון והאיבר הראשון הוא גדול מכולם אז נוסף חילוף על כל איבר אחר ונקבל 1+2+3….+n-1+n

כרצוי

לכן אם נציב

7998000

31996000

Search:

חסם עליון:

בהכנסה ה-i אנחנו מכניסים את האיבר המינימלי בעץ החדש, מכיוון והוא קטן מהשורש והמקסימלי גדול מהשורש, המרחק ביניהם יהיה המרחק בין השורש למקסימלי ועוד המרחק בין השורש למינימלי, וכל אחד מאלה חסום מלמטה ע"י log(i)/2 לכן מרחק יהיה חסום מלמעלה ע"י log(I) ולכן סך ההכנסות יהיה :

[ראינו במבוא מורחב למדמ"ח את הצעד האחרון]

חסם תחתון:

באותו אופן, ממקודם אנחנו יודעים שהמרחק בין החדש למקסימלי הוא לפחות העומק של העץ, שזה לכל היותר log(i) ולכן נקבל:

[ראינו במבוא מורחב למדמ"ח את הצעד האחרון]

ג. Chart, scatter chart

Description automatically generated

על מנת לבדוק האם התוצאות האימפריות מתאימות לתחזית, ביצענו ליניארזציה לערכים. אם התוצאות האימפריות באמת מתאימות נצפה לקבל קו מגמה ליניארי, וכך קרה. לכן התוצאות מתאימות לתחזית בסעיף ב'.

ד. נשתמש ברמז, נסמן לכל i את מספר האיברים הגדולים ממנו המופיעים לפניו ברשימה, נשים לב כי מהגדרה. בכל הכנסה של איבר i כזה המרחק בינו לבין המקסימום הוא לכל היותר O(log()), אכן: מכיוון וזהו עץ AVL קיים קבוע a כך שלאחר alog() עליות נהיה בתת עץ בגודל לפחות כך שהמקסימום הוא העלה הימני שלו, כל האיברים בתת עץ זה הם הקודמים לו, כלומר כל צומת שלא בתת עץ זה קטן מכל הצמתים בתת העץ, לכן i יצטרך להיכנס בתת עץ זה, הוא עלה בעת ההכנסה, ולכן עומקו בתת עץ זה יהיה O(log()) ולכן סך העליות והירידות יהיה O(log()+1) ולכן סך המרחקים יהיה:

וניזכר שזה סיבוכיות עלות המיון כפי שהוגדר בסעיף א'.

שאלה 2

א.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר סידורי i | עלות Join ממוצע לספליט אקראי | עלות ג'וין מקסימלי לספליט אקראי | עלות ג'וין ממוצע לספליט של האיבר המקסימלי בתת עץ השמאלי | עלות ג'וין מקסימלי לספליט של האיבר המקסימלי בתת עץ השמאלי |
| 1 | 3.1428 | 6 | 2.8888 | 13 |
| 2 | 2.7272 | 7 | 2.25 | 14 |
| 3 | 2.9090 | 8 | 2.5 | 15 |
| 4 | 2.5384 | 5 | 2.8333 | 17 |
| 5 | 2.3333 | 6 | 2.5714 | 18 |
| 6 | 2.6 | 6 | 2.5625 | 19 |
| 7 | 2.6 | 6 | 2.5882 | 20 |
| 8 | 2.4705 | 7 | 2.7058 | 22 |
| 9 | 2.3684 | 5 | 2.2631 | 22 |
| 10 | 2.815 | 5 | 2.65 | 24 |

ב. בתרחיש שמתחילים מהפרדססור של השורש:

בכל הצעדים הראשונים נבצע join בין עץ המפתחות הקטנים עד כה לבין הבן השמאלי של הצומת הנוכחי. כפי שיוסבר בסעיף ג, עלות כל פעולה כזו חסומה בין 1 ל 3, וישנן θ(log(n)) פעולות join כאלה, מתכונות AVL. לבסוף, הjoin האחרון מאחדת עץ ריק עם כל תת העץ הימני של השורש שגודלו θ(log(n)) ולכן עלותה θ(log(n)). לכן יש סה"כ θ(log(n)) פעולות. מהגדרת תטא ניתן לחסום עם קבועים  את כמות האיחודים בעץ ובאותו אופן ניתן לחסום עם קבועים את העלות של הjoin האחרון . ולכן נקבל עבור join ממוצע:

*ולכן סה"כ נקבל שסיבוכיות* של join ממוצע היא θ(1).

*בתרחיש שמתחילים מאיבר רנדומלי:*

נשים לב כי ישנם θ(log(n)) *join ים. בכל join אנחנו באים או מהבן הימני או מהבן השמאלי, ונגדיר פונקציית פוטנציאל f: לכל פעולת join i , נסמן ב f(i) את אורך רצף הjoin ים מאותו כיוון (joinים רצופים בכיוון הזה) כפול 2. נראה שכל פעולת join היא באמורטייז* θ(1)*. נפריד לשני מקרים: האם הjoin ממשיך ברצף הנוכחי או שובר אותו?*

*אם הjoin ממשיך את הרצף הנוכחי, נראה בסעיף הבא שהעלות חסומה בין 1 לבין 3 ולכן עלות הפעולה היא* θ(1).

*אם הjoin שובר את הרצף, נניח בה"כ שהרצף שנשבר היה של עליות שמאלה, וכעת עלינו ימינה, כלומר מאחדים את תת העץ של הבן הימני עם כל האיברים שאספנו עד כה בעץ של הגדולים מהצומת עליו עושים split. נסמן בh את הגובה של הצומת האחרון בו עשינו join ימינה (אם אין כזה, h=0). נשים לב שמכיוון שמדובר בעת AVL הגובה של הבן הימני של הצומת הנוכחי שעושים עליו join הוא לכל היותר f(i) +h. נסביר זאת: מתכונות AVL, בתרחיש קיצוני (לכל היותר) גובה כל צומת בסדרה גדול ב2 מהגובה של בנו ולכן הגובה של העץ הנוכחי גבוה לכל היותר בf(i) מגובהו של h. מכיוון וגובהו של הבן הימני של הצומת שגובהו h הוא h-1 או h-2 , נשים לב שעץ כל המפתחות שגדולים מהאיבר עליו עשינו split הוא איחוד של הבן הזה עם איזשהו עץ שהיה לפניכן ולכן גובהו לפחות h-2. לכן, עלות פעולת האיחוד הנוכחית חסומה מלמעלה על ידי f(i)+3 (הגובה האפשרי הכי גדול פחות ההכי קטן +1). נשים לב שf(i+1) = 1 כי הוא מתחיל רצף חדש.*

*ולכן פעולת הjoin הממוצעת במקרה הממוצע היא באמורטייז* θ(1).

ג. נשים לב שהצומת המדובר הוא הpredecessor של השורש ובפרט אין לו בן ימני. לכן, בפעולת הsplit בכל העליות למעט העליה האחרונה (לשורש) נעלה מצד ימין (כלומר הצומת הקודם בjoin הוא הבן הימני של הצומת הנוכחי).

נסביר מדוע כל הjoin ים האלה קטנים: נסמן את הגובה של הצומת הקודם בh. נוכיח באינדוקציה על h שגובה העץ של המפתחות הקטנים עד עכשיו (אחרי שאיחדנו עם h והתת עץ השמאלי שלו) הוא בין h לh-2.

בסיס: כאשר h=1 או h=0 זה המקרה שבו אנחנו בפרדססור עצמו. בשני המקרים האיחוד הוא עלה או צומת אונארי ולכן בשני המקרים האיחוד יצא תת העץ שהפרדססור הוא השורש וזה עץ בגובה 1 או 0 וזה מתאים למה שרוצים להוכיח.

צעד: נניח נכונות לh ונוכיח לאבא של h שגובהו הוא h+1 או h+2. נפרד למקרים לפי הגובה.

h+1: גובה תת העץ השמאלי הוא h או h-1 מתכונות AVL ולפי הנחת האינדוקציה נקבל שתת העץ שהצטבר עד כה הוא בין h לh-2. לכן האיחוד ביניהם יהיה בגובה h+1 לכל היותר ולכל הפחות h-1 כנדרש.

h+2: גובה תת העץ השמאלי הוא h או h+1מתכונות AVL ולפי הנחת האינדוקציה נקבל שתת העץ שהצטבר עד כה הוא בין h לh-2. לכן האיחוד ביניהם יהיה בגובה h+2 לכל היותר ולכל הפחות h כנדרש.

לכן הראנו באינדוקציה שכל הjoin ים הללו בעלות חסומה על ידי 3 תמיד, ולכן הjoin ים האלה חסומים בעלותם על ידי קבוע וזניחים ביחס למקסימום. נראה שהjoin הנותר הוא המקסימלי: הjoin הנותר הוא בין עץ ריק (תת העץ של "הגדולים" מהפרדססור אשר נשאר ריק כי המשכנו רק לעלות שמאלה כל פעם כאמור), לבין כל תת העץ הימני של השורש, שהרי הפעם עלינו דווקא ימינה. גודל תת העץ של השורש הוא לפחות גובה העץ פחות 2 ולכן עלות האיחוד בינו לבין עץ ריק יהיה גובה העץ פחות 1 שזה o(log(n)) כאשר n מספר הצמתים בעץ. n בסדר גודל של אלפים ומעלה ובפרט הגובה גדול בהרבה מ 3 ולכן הjoin האחרון הזה הוא אכן המקסימלי ועלותו o(log(n)).