סעיף א':

פעולת הכנסה לערימת פיבונאצ'י במימוש הינה מסיבוכיות  *. לכן בהתחלה סכום כל ה-1+m הכנסות עולות לנו*

פעולת delete-min() היחידה במימוש הינה מסיבוכיות:  *כ מספר הצמתים הנוכחי בערימה.*

פעולת Decrease-key מערימת פיבונאצ'י במימוש הינה מסיבוכיות *בamortized- . לכן סכום כל ה-* של Decrease-key *עולות לנו סה"כ עלות הסדרה*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Potential | totalCuts | totalLinks | Run-Time (ms) | m |
| 14 | 5 | 31 | 2 |  |
| 29 | 15 | 1054 | 1 |  |
| 44 | 30 | 33821 | 13 |  |
| 59 | 50 | 1082396 | 137 |  |

סעיף ב':

סעיף ג':

נראה כמה פעולות Link במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

את פעולת ה- Linkאנו מבצעים רק כאשר אנו עושים delete-min בשלב ה-consolidate. מכיוון שאנחנו לא מבצעים פעולות Decrease-key אז נתייחס לעצים בערימה כעצים בינומיים לאורך הניתוח. בהינתן שמספר ההכנסות היה m+1, נספור כמה ביצענו לאחר המחיקה: נשים לב כי מכיוון שמדובר בכמות ערימות בודדות מדרגה 0 אחרי המחיקה אז בעת consolidate אנחנו נאחד את הצמתים לעץ בינומי אחד.

עץ בינומי מדרגה k מצריך כמות חיבורים, מתקיים

, נוכיח באינדוקציה :

בסיס:

*צעד:*

סה"כ הכמות של ה- Linkשאנו מבצעים היא

נראה כמה פעולות Cut במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

פעולות Cut נעשות רק במקרה של Decrease-key. אנחנו עושים logm פעולות Decrease-key ומתקיים כי המיקום שבו אנחנו עושים Decrease-key דואג לכך שאנו לא חותכים מאב אחד 2 בנים (כל אב שונה) ומבצעים Cascading-Cuts. כלומר אנחנו נשארים עם logm כמות של פעולות Cut.

נראה מהו הפוטנציאל כתלות ב-m בסוף סדרת הפעולות:

נזכור כי הפוטנציאל מחושב ע''י:*. כפי שצוין קודם אנו מבצעים* logmפעולות Cut *עם כל פעולה אנחנו מוסיפים עץ לערימה (בערימה כפי שציינו יש עץ אחד) כי עשינו הפחתה של ערך המפתח למספר שלילי קטן מ – 1- ולכן נוסף לנו* logm+1 לפוטנציאל*.*

*מכיוון אנו מבצעים* logmפעולות Cut *לאבות שונים אז אנחנו גם נסמן את ה-* logm-1 *אבות, מכיוון שאחד האבות, הצומת שערכו -1 (עשינו* Decrease-key *לצומת עם ערך מפתח אפס) הוא בהכרח שורש לפי סדר ההכנסה ותהליך ה-* consolidate*.*

סה"כ*:*

סעיף ד':

נראה כמה פעולות Link במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

זהה לסעיף קודם סה"כ הכמות של ה- Linkשאנו מבצעים היא

נראה כמה פעולות Cut במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

במקרה זה Decrease-key פועל על צמתים שהם צאצאים אחד של שני ומתחיל בסדר מהשורש לכן אנו לא נצטרך לעשות сut כי כלל הערימה משתמר. סה"כ 0.

נראה מהו הפוטנציאל כתלות ב-m בסוף סדרת הפעולות:

מתבצע תהליך consolidate ולכן אנחנו נותרים עם עץ אחד. כפי שתיארנו בחלק הקודם - עקב הסדר לא נצטרך לעשות сut על מנת ליצור עץ חדש – כלומר כמות העצים אינה מתווספת וכך גם כמות המסומנים שנשארת 0.

סה"כ*.*

סעיף ה':

נראה כמה פעולות Link במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

את פעולת ה- Linkאנו מבצעים רק כאשר אנו עושים delete-min ולכן בגרסא הזאת לא מתבצע פעולת Link. סה"כ הכמות של ה- Linkשאנו מבצעים היא .

נראה כמה פעולות Cut במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

פעולות Cut נעשות רק בביצוע Decrease-key במקרה שהמפתח לא מקיים את כלל הערמה. מכיוון שמדובר בעצים שכולם דרגה 0 אז לא מתבצע פעולת Cut. סה"כ 0.

נראה מהו הפוטנציאל כתלות ב-m בסוף סדרת הפעולות:

לא מתבצע תהליך consolidate ולכן אנחנו נותרים עם m עצים מדרגה 0. מכיוון שהם שורשים אז אין מה לסמן בעת Decrease-key. סה"כ*.*

סעיף ו':

נראה כמה פעולות Link במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

זהה לסעיפים ג,ד. סה"כ הכמות של ה- Linkשאנו מבצעים היא

נראה כמה פעולות Cut במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

בלי להתחשב בהוספה האחרונה - זהה לסעיף ג. סה"כ אנחנו מקבלים עם logm כמות של פעולות Cut. עכשיו נחשב התרומה של ההוספה – אנחנו בעצם מבצעים את הפעולה של Decrease-key על צומת-עלה של ענף שלם של אבות מסומנים (שפעלנו עליהם לפני) עד השורש (לא כולל) כלומר כמות של צמתים מסומנים, ולכן בעת הפעולה אנחנו נבצע תהליך Cascading-Cuts עד השורש לכן נוסיף עוד  *לסכום. סה"כ .*

נראה מהו הפוטנציאל כתלות ב-m בסוף סדרת הפעולות:

עקב תהליך ה- Cascading-Cuts *שתיארנו לפני, נוצרים לנו עצים חדשים שנוספים לכמות העצים שהזכרנו בסעיף ג ולכן*

*סה"כ עצים. אחרי התהליך של כל* Cascading-Cuts*, כל האבות המסומנים הפכו לשורשים משמע אין מסומנים.*

*סה"כ*

עלות הכי יקרה:

עקב תהליך ה- Cascading-Cuts *שתיארנו לפני, העלות הכי יקרה היא .*

טבלה מסכמת:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **decreaseKey max cost** | **Potential** | **totalCuts** | **totalLinks** | **case** |
|  |  |  |  | ג |
|  |  |  |  | ד |
|  |  |  |  | ה |
|  |  |  |  | ו |

סעיף א':

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Potential | totalCuts | totalLinks | Run-Time (ms) | m |
| 6 | 0 | 723 | 3 | 728 |
| 6 | 0 | 7278 | 3 | 6560 |
| 9 | 0 | 66318 | 41 | 59048 |
| 10 | 0 | 597749 | 226 | 531440 |
| 14 | 0 | 5380704 | 2759 | 4782968 |

סעיף ב':

פעולת הכנסה לערימת פיבונאצ'י במימוש הינה מסיבוכיות:  *ב-WC*

פעולת delete-min() במימוש הינה מסיבוכיות:  *ב-WCכאשר n הינו מספר הצמתים בערימה.*

אזי:

סעיף ג':

נראה כמה פעולות Link במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

את פעולת ה- Linkאנו מבצעים כאשר אנו עושים delete-min בשלב ה-consolidate. מכיוון שאנחנו לא מבצעים פעולת Decrease-key אז נתייחס לעצים בערימה כעצים בינומיים לאורך הניתוח. בהינתן שמספר ההכנסות היה m+1, נספור כמה ביצענו לאחר המחיקה הראשונה:

עץ בינומי מדרגה k מצריך כמות חיבורים, נסמנה ,

מתקיים כלומר . על מנת לחשב אילו עצים מורכבת הערימה נשתמש בשיטה דומה לחישוב מה המספר בייצוג הבינארי של m-1:

עכשיו לאחר פעולת ה- consolidate נותרנו עם כמות עצים כמספר הביטים 1 בייצוג הבינארי של מספר m-1 וכל העצים שנותרנו איתם הם עצים בינומיים בדרגות שונות. בנוסף עקב סדר הכנסת האיברים ופעולת ה- consolidate אנחנו מקבלים כי העץ עם הדרגה הקטנה ביותר, נסמנה d, הוא בעל מפתחות יותר קטנים מכל עץ בדרגה גדולה יותר (האיבר שנכנס ראשון "יצא אחרון" בתהליך ה- consolidate).

אם נבצע עוד מחיקה אז מתובנות קודמות אנחנו נמחק שורש מהעץ בדרגה הכי קטנה, לאחר המחיקה מתכונה ערימה בינומית יוספו לערימה עצים שונים בדרגות 0 עד d-1 שהם בהכרח מדרגה יותר קטנה משאר העצים. לכן אין בערימה אף עץ מאותה דרגה ולכן לא נבצע אף Link במהלך ה- consolidate. נשים לב כי שוב מאופן הכנסת האיברים בהתחלה, העץ עם השורש המינימום יהיה העץ הכי קטן והוא יהיה שורש יחיד – עץ בדרגה 0.

התהליך הזה חוזר על עצמו: אנחנו נמחק שורש של עץ עם הדרגה הכי קטנה כי אצלו המפתח המינימלי ונפרק אותו לעצים מדרגה קטנה יותר, ומכיוון שהוא היה בעל הדרגה הקטנה ביותר לפני מחיקה מובטח לנו שאחרי הוספת ילדיו לא יהיו עצים עם דרגות תואמות בערימה ולכן שוב לא מתבצע Link בתהליך ה- consolidate.

לכן סה"כ הפעולות שביצענו הם הכמות שביצענו לאחר המחיקה הראשונה :

נראה כמה פעולות Cut במדויק מתבצעות במהלך סדרת הפעולות כתלות ב-m:

פעולות Cut נעשות רק במקרה של Decrease-key ולא נעשות בפעולות אילו כלל לכן מספר הפעולות הינו 0.

נראה מהו הפוטנציאל כתלות ב-m בסוף סדרת הפעולות:

נזכור כי הפוטנציאל מחושב ע''י:*. נשים לב כי אנו לא מבצעים* פעולות Cut *לכן אף צומת לא תהיה מסומנת. נחשב את מספר העצים ונסיים. מספר העצים בערימה יהיה כמספר ה-"1"-ים בייצוג הבינארי של . נסביר תחילה מדוע : התחלנו עם צמתים לאחר מכן מחקנו לכן נותרו כ- צמתים. נסביר כעת למה מספר ה-"1"-ים בייצוג הבינארי: לאחר המחיקה הראשונה נשארנו עם מספר עצים בהתאם לייצוג הבינארי של , בכל מחיקה נשאר עם צומת אחת פחות על כן מספר העצים בערימה יהיה כמספר ביטי 1 בייצוג הבינארי של מספר הצמתים.*

*אזי:*