מבני נתונים 2 - הרצאה מס׳ 2

Lazy Meld וערימות פיבונצ׳י

הקדמה

בשיעור הקודם ראינו מימוש של ערימה שתומכת במיזוג, בעזרת רשימה מקושרת של עצים בינומיאליים. כל הפעולות נעשו ב $O(\log n)$ נרצה לממש את הערימה באופן שונה כך שכל הפעולות למעט *del_min* יהיו $O(1)$ - זמן משוערך .

Lazy Meld

בכל רגע נחזיק רשימה מקושרת של עצים בינומיאליים מסודרים בסדר ערימה, אבל יתכן שיהיה יותר מעותק מעץ בינומיאלי מסדר מסויים. נשמור מצביע לעץ שלשורשו המפתח המינימלי מבין המפתחות במבנה.

מימוש הפעולות:

make_heap(k) - יצירת עץ יחיד עם צומת אחד שמפתחו k . האיבר המינימלי הוא שורש עץ זה.

insert(k) - הוספת עץ עם צומת יחיד לרשימה המקושרת. עדכון המצביע לבן המינימלי.

find_min - החזרת המפתח המוצבע ע״י המצביע לשורש בעל המפתח המינימלי.

meld(h1,h2) - שרשור שתי הרשימות המקושרות של שתי הערימות. עדכון המצביע בהתאם למינימום מבין המפתחות המינימלים של שני היערות.

del_min - ראשית ניקח את הצומת שמכיל את המפתח המינימלי ונמחוק אותו. העץ הבינומיאלי שמפתח זה היווה את שורשו, מתפרק עכשיו לתת עצים, כל אחד מהם בינומיאלי. נצרף עצים אלה לרשימה המקושרת הקיימת. עכשיו יש צורך למצוא את השורש בעל המפתח המינימלי מבין כל העצים שביער. מכיוון שזה מצריך מעבר על כל העצים בערימה, ננצל מעבר זה לביצוע *meld* .
ראשית נמיין את כל העצים לפי דרגתם (מספר הבנים של השורש), בעזרת *bucket sort*. עכשיו נעבור על כל העצים באותה דרגה (באותו *bucket*), החל מהדרגה הנמוכה ביותר, ונאחד כל שניים תוך שמירה על סדר ערימה ונקבל עץ בינומיאלי חדש מדרגתו גדולה באחד. נמשיך עד שנסיים לעבור על כל התאים (יתכן ויווצרו תאים חדשים). לבסוף נקבל לכל היותר עץ בינומיאלי אחד מכל דרגה. תוך כדי המעבר נמצא גם את השורש בעל המפתח המינימלי ונעדכן את המצביע אליו.

סיבוכיות:

זמן אמיתי:

קל לראות שארבע הפעולות הראשונות לוקחות $O(1)$.

הפעולה האחרונה מצריכה $O(1)$ מעברים על כל העצים לצורך מיון, איחוד ומציאת המינימום, ובנוסף טיפול במערך הדליים - $O(\log n)$, לכן סה״כ $O(t+\log n)$, כאשר t הוא מספר העצים בערימה לפני המחיקה (מיד אחרי המחיקה מספר זה יכול לגדול ב $\log n$ לכל היותר).

זמן משוערך:

נתייחס למבנה הנתונים כמכיל מספר ערימות (כל פעולה מתייחסת לאחת מהערימות). פונקציית הפוטנציאל תהיה סה"כ מס' העצים בכל הערימות.
כל אחת מארבע הפעולות הראשונות גורמות להוספה של עץ אחד לאחת הערמות לכל היותר, כלומר $O(1)$ שינוי לפוטנציאל ולכן זמן משוערך שלהן : $a = t + \Delta\varphi = O(1)$.
פעולת מחיקה, גורמת למס' העצים לקטון מ t ל $\log n$ לכל היותר, ולכן

$$\Delta\varphi = \varphi(\text{after}) - \varphi(\text{before}) \le \log n - t \ .$$

לכן,

$$a = t + \Delta\varphi \ \le \ t + \log n + \log n - t \ = O(\log n)$$

Fibonacci Heaps

נוסיף פעולה חדשה : ***decrease (x,Δ)*** - הורדת ערכו של המפתח של ***x*** ב Δ .
<u>מימוש</u> : נתק את תת העץ של ***x*** מאביו, והוסף את העץ החדש לרשימה המקושרת של הערימה. בעץ החדש הורד את ערך המפתח של ***x*** ב Δ . בנוסף, כל צומת שמאבד שני בנים, מתאבד, כלומר מתנתק מאביו.

<u>נשים לב</u> : כל עוד לא ביצענו פעולה זו, המצב הוא בדיוק כמו ב ***lazy meld*** , כלומר יש לנו רשימה מקושרת של עצים בינומיאלים. ברגע שנתחיל לבצע ניתוקים, הערימה תכיל גם עצים שהם לא בינומיאלים.

<u>שאר הפעולות</u> : כמו ב ***lazy meld*** . מחיקת המינימלי תעשה באותו אופן, כאשר העצים בערימה יסודרו לפי דרגתם ויחוברו עצים בעלי דרגת שורש שווה לקבלת עץ בדרגת גדולה באחד.
נוסיף גם את פעולת ***delete(x)*** למחיקת איבר ***x*** מהערימה. נממש פעולה זו ע"י ***decrease(x,∞)*** ואח"כ ***del_min()*** .

<u>סיבוכיות</u>

לצורך האנליזה נצטרך לדעת מס' עובדות על סדרת פיבונצ'י :

הגדרה : $F_0 = 0 \quad F_1 = 1 \quad F_{n+1} = F_n + F_{n-1}$

תכונות :

א. $F_{n+2} = 2 + \sum_{i=2}^{n} F_i$

ב. נוסחה סגורה לסדרת פיבונצ'י : $F_n = \frac{1}{\sqrt{5}}\left(\Phi^n - \overline{\Phi}^n\right)$

כאשר : $\Phi = \frac{1+\sqrt{5}}{2} = 1.608 \qquad \overline{\Phi} = \frac{1-\sqrt{5}}{2} = -0.688$

מכוון ש $|\Phi| > 1$, $|\overline{\Phi}| < 1$ מתקיים : $F_n \approx \frac{1}{\sqrt{5}}\Phi^n$

נרצה לחסום את מספר העצים בערימה שלאחר ביצוע מחיקה, וכן את מספר העצים שנוצרים כאשר מוחקים שורש של עץ ב $O(\log n)$. כדי שנוכל לעשות זאת, נראה שמספר הצמתים בתת-עץ כלשהו בערימת פיבונצי' הוא אקספוננציאלי בדרגת העץ.

למה 1: בערימת פיבונצי' לבן ה - ***i*** של צומת ***x*** יש דרגה לפחות ***i - 2*** .

<u>הוכחה:</u>

יהי ***y*** הבן ה ***i*** של ***x*** . נניח שבזמן הצטרפותו של ***y*** לעץ הוא הבן ה ***d + 1*** של ***y*** . אז הדרגה של ***y*** ברגע ההצטרפות היא ***d*** , כי מאחדים עצים עם דרגה שווה. לא יתכן שמאז ***y*** איבד יותר מבן אחד, כי אחרת הוא היה מתנתק מאביו, לכן כרגע יש ל ***y*** לפחות ***d - 1*** בנים. מצד שני, כל בן חדש שנוסף ל ***x*** , נוסף אחרי ***y*** , ולכן:

$$i \le d+1$$
$$\deg(y) \ge d-1 \ge i-2$$

למה 2: בערימת פיבונצי' לצומת מדרגה ***k*** יש לפחות F_{k+2} צאצאים.

<u>הוכחה:</u>

נסמן ב S_k את מס' הצמתים בתת-העץ המינימלי (מבחינת מס' הצמתים) שדרגת שורשו ***k*** , שיכול להיווצר בערימת פיבונצי'. קל לראות שסדרה זו מונוטונית עולה ב ***k*** (לכל עץ שדרגתו ***k + 1*** , קיים עץ שדרגתו ***k*** , ומכיל פחות צמתים - זה המתקבל ע"י ביצוע ***decrease()*** , לאחד מבניו של שורש תת-העץ). מספיק להראות ש $S_k \ge F_{k+2}$.

<u>באינדוקציה:</u>

<u>S_0:</u> בעץ שדרגתו 0 יש צומת אחת בדיוק, ולכן: $S_0 = 1 = F_2$.

<u>S_1:</u> בעץ שדרגתו 1 יש לפחות שני צמתים (השורש של תת העץ, ובנו האחד), ולכן $S_k \ge 2 = F_3$.

<u>צעד:</u> נתבונן בעץ המינימלי ***T*** האפשרי שדרגתו ***k*** בערימות פיבונצי'. נסמן ב ***size(v)*** את מספר הצמתים בתת העץ ששורשו ***v*** . מתקיים:

$$|T| = 1 + \sum_{i=1}^{k} size(son_i) \quad \ge \quad 2 + \sum_{i=2}^{k} size(son_i)$$

מלמה 1 מתקיים:

$$\deg(son_i) \ge i-2$$

ולכן, בגלל מונוטוניות:

$$size(son_i) \ge S_{i-2}$$

מהנחת האינדוקציה מתקיים:

$$S_{i-2} \ge F_i$$

ולכן נקבל:

$$|T| \ge 2 + \sum_{i=2}^{k} F_i \quad = \quad F_{k+2}$$

משי"ל.
מסקנה:

מס' הצמתים בתת-עץ ששורשו בדרגה k הוא $\Omega(\Phi^k)$ (נובע מכך שהמספר הפיבונצ'י ה-k-י מקיים: $F_k = \Theta(\Phi^k)$).

לכן הדרגה המקסימלית במבנה שמכיל ערימות פיבונצ'י עם סה"כ n צמתים היא: $O(\log_\Phi n) = O(\log n)$.

ניקח פונקציית פוטנציאל שעבור מבנה שמכיל מספר ערימות פיבונצ'י ערכה הוא מספר העצים + פעמיים מס' הצמתים השכולים במבנה (שורשים של עצים בערימה אף פעם לא שכולים, וזה אומר שצומת שהתאבד כבר לא שכול יותר).
באופן דומה לניתוח עבור ***lazy meld*** לארבע הפעולות הראשונות יש זמן משוערך $O(1)$.

עבור פעולת ***del_min*** הניתוח הוא בדיוק כמו קודם, כי אין שינוי במספר הצמתים השכולים, כלומר $O(\log n)$ זמן משוערך.

בפעולת ***decrease*** יתכן ונוצרה קסקדה של התאבדויות. נניח קרו l התנתקויות. אז פרט לצומת הראשון, $l-1$ צמתים שכולים התאבדו, ולכן הם יותר כבר לא שכולים. הצומת האחרון שהתאבד הפך את אביו לשכול, ולכן צומת אחד הפך שכול. סה"כ השינוי במס' השכולים הוא: $-(l-1)+1 = -(l-2)$. מס' העצים החדשים שנוצרו הוא l. והזמן האמיתי שערכה הפעולה הוא $O(l)$. סה"כ מקבלים:

$$a = t + \Delta\varphi \quad = \quad O(l) + l - 2(l-2) \quad = \quad O(l) - l + 4 \quad = \quad O(1)$$

פעולת ***delete(x)*** מורכבת מפעולת ***decrease*** ופעולת ***del_min*** ולכן הזמן המשוערך שלה הוא $O(\log n)$

סיכום

מימשנו את הפעולות ***make_heap*** , ***insert*** , ***find_min*** , ***meld*** , ***decrease*** ב $O(1)$ זמן משוערך ואת הפעולה ***del_min*** ב $O(\log n)$, כאשר n הוא מספר המפתחות במבנה ברגע נתון. מתחילים ממנה שמכיל 0 ערימות, והניתוח מדבר על מבנה שמכיל מספר ערימות.