שיטות מיון וחיפוש, סיכום

30/1/01

1. **חיפוש איבר x במערך לא ממוין**

(א) רעיון: נעבור אחד אחד על אברי המערך (לרוב ע"י לולאה) ונבדוק, עבור כל אחד, האם שווה ל x. אם כן אז סיימנו, אם לא אז ממשיכים לאיבר הבא. אם בדקנו את כל האיברים סימן ש- x לא נמצא במערך.

(ב) סיבוכיות: במקרה הגרוע נעבור על כל אברי המערך ולכן $O(n)$. במקרה הטוב נמצא "על ההתחלה" ולכן $\Omega(1)$.

2. **חיפוש איבר x במערך ממוין**

(א) רעיון: נחזיק שני אינדקסים/מצביעים `l, r`. `l` יצביע על האיבר השמאלי (הקטן ביותר) של תת-המערך שמסתכלים עליו כרגע ובדומה `r` יצביע על האיבר הימני (הגדול ביותר) של תת-המערך שמסתכלים עליו כרגע. במהלך ריצת האלגוריתם נדאג לכך שאם האיבר קיים במערך אז הוא קיים גם בתת-המערך שאנו מסתכלים עליו כרגע. נחלק למקרים:

i. אם `l>r` אז מסתכלים בעצם על מערך ריק ולכן נפסיק, לא מצאנו.

ii. אחרת, נחשב `t=(l+r)/2` (האיבר האמצעי בין `l` ו `r`). נבדוק את `a[t]` ביחס ל x, מקרים:

A. `a[t]=x` ואז מצאנו, נפסיק.

B. `a[t]>x` ואז אם האיבר קיים אז הוא חייב להיות אחד מבין `a[l],..,a[t-1]`. נבצע `r=t-1` ונמשיך.

C. `a[t]<x`, זה המקרה הסימטרי לקודם ולכן נבצע `l=t+1` ונמשיך.

(ב) סיבוכיות: בכל איטרציה גודל המערך שמסתכלים עליו יורד בלפחות חצי ולכן $O(\log n)$. יכול להיות שנמצא את האיבר "על ההתחלה" ולכן $\Omega(1)$.

3. **מיון bubble sort**

(א) רעיון: נבצע $n-1$ איטרציות, באיטרציה ה- `i` `(i=1,..,n-1)` נדאג שבסוף האיטרציה האיבר במקום ה `a[n-i]` יהיה האיבר הנכון. נדאג לכך ע"י זה שנעבור על זוגות איברים צמודים במערך (משמאל לימין), אם המספרים בתוך הזוג אינם בסדר הנכון נחליף אותם. פעולה זו "תבעבע" את האיבר המתאים למקום ה `a[n-i]`.

(ב) סיבוכיות: מתבצעות $n-1$ איטרציות. באיטרציה ה- `i` מתבצעות $\Theta(i)$ פעולות. לכן סה"כ זמן הביצוע הוא $\Theta(n^2)$. האלגוריתם דורש מקום נוסף $O(1)$.

4. מיון **merge sort**

(א) רעיון: נבצע $\Theta(\log n)$ איטרציות, נדאג לכך שבתחילת האיטרציה ה-i ‏(i=1,..) המערך יורכב ממקטעים ממוינים באורך של 2^{i-1} (חוץ מה-מקטע האחרון שאולי יהיה קצר יותר) ובסופה יורכב ממקטעים ממוינים באורך של 2^i (חוץ מהמקטע האחרון שאולי יהיה קצר יותר). נהפוך שני מקטעים סמוכים למקטע אחד ע"י פעולת מיזוג.

(ב) סיבוכיות: פעולת מיזוג של שני מערכים באורכים k,l לוקחת $\Theta(k+l)$. נשים לב שבכל איטרציה כל איבר משתתף בפעולת מיזוג אחת לכל היותר, וכן לפחות מחצית מהאיברים משתתפים בפעולת מיזוג. לכן כל איטרציה לוקחת $\Theta(n)$. יש $\Theta(\log n)$ איטרציות ולכן סה"כ זמן הריצה הוא $\Theta(n \log n)$. האלגוריתם דורש מערך עזר של n מקומות, לכן סיבוכיות המקום הנוסף היא $\Theta(n)$.

5. מיון **quick sort**

(א) רעיון: נבחר אקראית איבר במערך ונסדר את המערך כך שכל האיברים שקטנים מאיבר זה יהיו משמאלו וכל האיברים שגדולים ממנו יהיו מימינו. נמיין רקורסיבית את שני תתי-המערכים שייצרנו. בגמר פעולה זאת המערך כולו יהיה ממוין.

(ב) סיבוכיות: במקרה הגרוע עלולה להיות $\Omega(n^2)$. כאשר כל האיברים שונים ואנו בוחרים איבר ציר בצורה אקראית הזמן הממוצע יהיה $O(n \log n)$. סיבוכיות המקום הנוסף היא $O(1)$. פרקטית כאשר רוצים למיין אז quick sort נותן את התוצאות הטובות ביותר.

6. מיון **bucket sort**

(א) רעיון: אם תחום הערכים של אברי המערך הוא M ניתן לסרוק את המערך ולספור (ע"י שימוש במערך עזר) כמה פעמים כל איבר הופיע. בעזרת אינ-פורמציה זו ניתן למיין את המערך: נכתוב בהתחלה את האיבר הקטן ביותר כמספר הפעמים שהופיע, אח"כ את המספר השני הכי קטן כמספר הפע-מים שהופיע וכו'.

(ב) סיבוכיות: שלב איפוס מערך העזר לוקח $\Theta(M)$, שלב סריקת המערך לוקח $\Theta(n)$, שלב כתיבת המערך מחדש לוקח $\Theta(M+n)$. לכן סה"כ זמן ריצה $\Theta(M+n)$ ומקום נוסף $\Theta(M)$.

(ג) הערה: אלגוריתמי החיפוש הקודמים שראינו לא היו תלויים בגודל התחום של אברי המערך (מיוני השוואות), באלגוריתם זה גודל התחום הוא מש-מעותי.

7. סיכום

(א) ראינו מיוני השוואה שלוקחים $O(n \log n)$ זמן (אופטימלי). ראינו מיוני הש-וואה שצריכים $O(1)$ זיכרון נוסף (אופטימלי).

(ב) לא ראינו אלגוריתם שמשלב את שתי התכונות הללו. האם קיים? כן! אלגוריתם שמשיג את שני החסמים הוא heap sort, נלמד בקורס מבני נתונים 1.