מבני נתונים 08a

תרגול 11
28/2/2008

מיון

ליאור שפירא

תרגיל 1

- בהינתן n מספרים שלמים בטווח $1..n^c$ כיצד ניתן למיין בזמן $O(n \cdot c)$
- פתרון
  - נתחיל ממספרים בטווח $1..n^2$ האם ניתן למיין אותם ב$O(n)$?
  - אילו אלגוריתמים אנו מכירים שממיינים בזמן ליניארי?
    - Counting sort – בהינתן n מספרים בטווח 0..k $\theta(k+n)$
    - $k=O(n)$
  - נכתוב כל מספר בבסיס n, כמה ספרות?
  - נמיין כמו radix sort, $O(n)$ לכל ספרה ולכן $2O(n)=O(n)$
  - עבור הבעיה המקורית כמה ספרות יהיו לנו?

חסמים עליונים ותחתונים

- חסם עליון: הוכחה לזמן ריצה מקסימלי עבור כל קלט. למשל: נוסחאות נסיגה, פונקציית פוטנציאל
- חסם תחתון: קלט לדוגמא עבורו האלגוריתם לוקח הרבה זמן.
- נניח שהוכחנו שאלגוריתם מסויים פותר את הבעיה לכל קלט בזמן $O(t)$
- אזי $O(t)$ הוא חסם עליון לזמן הנדרש לפתור את הבעיה WC

איך נמצא חסם הדוק לבעיה?

- נניח הוכחנו שעבור בעיה מסוימת

קיים אלגוריתם שפותר את הבעיה לכל קלט בגודל n, בזמן $O(\log n)$.

- כדי למצוא חסם תחתון לבעיה יש להוכיח ש:

לכל אלגוריתם שפותר את הבעיה קיים קלט שעבורו זמן הריצה הוא $\Omega(\log n)$.

זה לא אומר שאין אלגוריתם שפותר את הבעיה בזמן $O(1)$ עבור קלטים מסוימים!

חסמי זמן ריצה לבעיה

- חסם עליון: $O(t)$: אלגוריתם לדוגמא שפותר את הבעיה לכל קלט בזמן $O(t)$
- חסם תחתון: $\Omega(t)$:
  - מיפוי כל אלגוריתם לעץ החלטות בגובה $\Omega(t)$
  - רדוקציה לבעיה שהוכחנו בכיתה עם חסם תחתון על זמן הריצה (מיון או חיפוש בינארי במודל ההשוואות).
  - זמן קריאת הקלט. למשל: כל אלגוריתם לבעיית מציאת המינימום מ-n מספרים חייב לעבור על כל הקלט ולכן לוקח זמן $\Omega(n)$

חסם תחתון על מיון על ידי השוואות

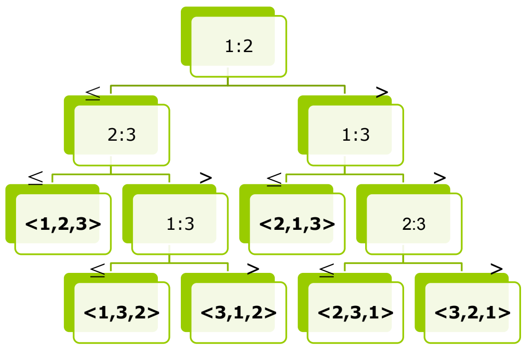
- אם אלגוריתם המיון שלנו מתבצע רק ע"י השוואות, כמה סידורים אפשריים יש ל-n אברים?
  - יש n! סידורים אפשריים
- For <a,b,c>
  - a,b,c
  - a,c,b
  - b,a,c
  - b,c,a
  - c,a,b
  - c,b,a

$\left.\begin{matrix}\\ \\ \\ \\ \\ \end{matrix}\right\} n!$

חסם תחתון על מיון על ידי השוואות

- ניתן לייצג כל אלגוריתם מיון (השוואתי) על ידי עץ החלטות בינארי
  - למה ניתן לייצג כעץ?
  - למה דווקא כעץ בינארי?
- בעץ עצמו
  - כל צומת מייצגת את הסידור החלקי כפי שידוע לנו עד נקודה זו
  - קשתות העץ הן תוצאות ההשוואות

חסם תחתון על מיון על ידי השוואות



חסם תחתון על מיון על ידי השוואות

- מס' ההשוואות המקסימאלי = עומק העלה העמוק ביותר בעץ
  - מס' ההשוואות הממוצע = עומק עלה ממוצע
- עץ החלטה למיין n אברים הוא בעל n! עלים בהכרח
  - עץ בינארי מעומק d הוא בעל 2^d עלים לכל היותר
    - עץ בינארי בעל 2^d עלים חייב להיות בעל עומק d
    - עץ בעל n! עלים חייב להיות מעומק של לפחות |log(n!)|
  - ולכן כל אלגוריתם מיון מבוסס השוואות דורש לפחות |log(n!)| השוואות במקרה הגרוע ביותר.

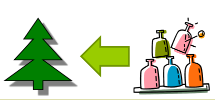
חסם תחתון על מיון על ידי השוואות

$$
\begin{aligned}
\log_2 n! &= \log(n(n-1)(n-2)\ldots(2)(1)) \\
&= \log n + \log(n-1) + \ldots + \log 2 + \log 1 \\
&\geq \log n + \log(n-1) + \ldots + \log(n/2) \\
&\geq n/2 \log n/2 \\
&= n/2 \log(n - n/2) \\
&= \Omega(n \log n)
\end{aligned}
$$

חסם תחתון על חיפוש

- מה החסם התחתון על חיפוש במערך ממוין?
  - במקרה הגרוע ביותר נשווה לכל איבר
  - ז"א בעץ יש n עלים
  - מה העומק המינימלי של עץ בינארי בעל n עלים?
    - זמן לוגריתמי O(logn)

תרגיל 2



- כמה "יקר" להפוך ערמה לעץ חיפוש בינארי?
- נראה שהחסם התחתון הוא nlogn
- נפתור ברדוקציה על דרך השלילה
  - נניח שניתן להפוך ערמה לעץ חיפוש בינארי ב-f(n)
  - בהינתן n מספרים, מיינו את המספרים (במודל ההשוואה)
    - נבנה ערמה בזמן לינארי (O(n))
    - נמיר את הערמה לעץ בינארי מאוזן בזמן f(n)
    - נעבור על העלים ב-in order ונדפיס אותם (ממוינים) בזמן לינארי
    - **סה"כ זמן – O(n) + f(n) + O(n)**
    - לכן, אם f(n)<<n·log(n) ניתן למיין n מספרים בזמן קטן מ- O(n·logn) ואנו יודעים שזה בלתי אפשרי. מ.ש.ל.

תרגיל 3

דוגמה:

j=6

1	2	4	5	6	9	**26**	15	14	3

- נתונים n אלמנטים במערך A. קיים אינדקס j כך שלכל $i<=j$ מתקיים $A[i]<A[i+1]$ ולכל $i>j$ מתקיים $A[i]>A[i+1]$
  1. איך ניתן למצוא את j?
  2. מדוע $\Omega(\log n)$ הוא חסם תחתון על מס' ההשוואות למציאת j?
- פתרון (1)
  - נפתור בדומה לחיפוש בינארי:
    1. מצא את האבר ה-n/2
    2. אם הוא גדול משני שכניו אזי מצאנו את j
    3. אחרת נמשיך בחצי המתאים רקורסיבית

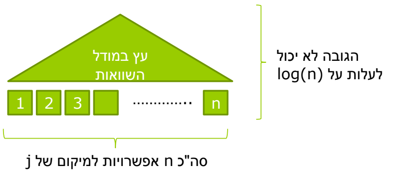
    O(logn)

טעות נפוצה: לחפש את החציון (למה לא טוב?)

תרגיל 3

2.מדוע $\Omega(\log n)$ הוא חסם תחתון על מס' ההשוואות למציאת j?

- פתרון (2)
  - אופציה א': ע"י עץ השוואות



  - אופציה ב' ע"י רדוקציה לחיפוש בינארי: נניח נתון מערך ממוין A ואבר אותו מחפשים X...

תרגיל 4

- הוכיחו כי לא יכול להיות אלגוריתם במודל ההשוואות בו עבור קלט באורך n, לפחות חצי מהפרמוטציות האפשריות של המספרים מ-1 עד n ניתנות למיון בזמן ליניארי.
  (עבור המחצית השניה של הפרמוטציות, לאלגוריתם מותר להחזיר כל דבר, אפילו סדר שגוי).

הסוף

שבוע נעים