מבנה נתונים ואלגוריתמים - הרצאה 2

1 בנובמבר 2011

תזכורת משיעור קודם

- נאמר ש$f(n)=O(g(n))$ אם עבור $n>N$ ו$c>0$ מתקיים:

  $f(n)\leq c\cdot g(n)$

- נאמר ש$f(n)=\Omega(g(n))$ אם עבור $n>N$ ו$c>0$ מתקיים:

  $f(n)\geq c\cdot g(n)$

- נאמר ש$f(n)=\Theta(g(n))$ אם $f(n)=O(g(n))$ וגם $f(n)=\Omega(g(n))$.
- הגדרנו קבוצת בעיות P - כל הבעיות להן קיים אלגוריתם שזמן החישוב שלו חסום ע"י פולינום - $f(n)=O(\text{Polynom})$.
- עבור בעיה בהינתן קלט בינרי הגדרנו שתי סוגי בעיות:
  1. בעיית פתרון - מה הוא הקלט שמקיים פתרון מסוים?
  2. בעיית בדיקה - מהו הפלט שנותן קלט נתון?
- הגדרנו קבוצת בעיות NP - בעיות אותן ניתן לבדוק (להפעיל בעיית בדיקה) בזמן שחסום ע"י פולינום.
  קבוצת בעיות NP שקולה לקבוצת הבעיות שניתן לפתור בזמן שחסום על ידי פולינום במחשב שיכול לפצל בעיות ולפתור אותן במקביל (מחשב לא דטרמיניסטי).
- הגדרנו רדוקציה בזמן פולינומיאלי: ניתן לעשות רדוקציה בזמן פולינומיאלי של בעיה A לבעיה B אם העלות של בעיה A היא פולינומיאלית, בהנחה שסופרים כל פתרון של B כפעולה אחת.
- בעיה היא שלמה - Complete בקבוצה מסוימת אם ניתן לעשות רדוקציה לכל הבעיות בקבוצה לבעיה הזו.
- בעיית SAT: בהינתן ביטוי לוגי, שמכיל את הפעולות $\sim$, $\vee$, $\wedge$ ו$()$ וסדרה של משתנים בינריים, האם קיים צירוף של ערכים של המשתנים הבינריים עבורו הביטוי ייתן 1.
- משפט Cook - בעיית SAT היא בעיית $NP-C$.

רקורסיה

מיון - יש לנו רשימת מספרים ואנו רוצים לסדר אותם מהגדול לקטן. נתבונן באלגוריתם שנקרא לו Slow-Sort:

1. $\ell=0$
2. $\text{Find } \max(A)$
3. $B(\ell)=\max(A)$
4. $\text{Remove } \max(A) \text{ from A}$

5. $\ell{+}{+}$
6. כל עוד A לא ריק, חזור ל3.
7. החזר B.

זמן הריצה של אלגוריתם זה הוא:

$$n+n-1+n-2+...+1=\Theta\left(n^{2}\right)$$

כעת נתבונן באלגוריתם שנקרא Quick-Sort:

1. אם גודל A הוא 1, החזר את A.
2. בחר ערך כלשהו בA, נסמן אותו x.
3. העבר את כל הערכים בA שקטנים מx לשמאלו ואת הערכים שגדולים או שווים לx לימינו.
4. בצע Quick-Sort למערך עד x וQuick-Sort למערך מx.
5. החזר את המערך הממוין.

למשל, ניקח את המערך:

$$7,\ 12,\ -1,\ 3,\ 27,\ 12,\ 91,\ -125,\ 3,\ 2,\ 2,\ 17,\ 8$$

נבחר את 7 כx, נקבל את המערך:

$$-1,\ 3,\ -125,\ 3,\ 2,\ 2,\ 7,\ 12,\ 27,\ 12,\ 91,\ 17,\ 8$$

כעת צריך להפעיל Quick-Sort על שני המערכים הבאים:

$$-1,\ 3,\ -125,\ 3,\ 2,\ 2$$
$$12,\ 27,\ 12,\ 91,\ 17,\ 8$$

נעשה על המערך הראשון, נבחר את 1- כx:

$$-125,\ -1,\ 3,\ 3,\ 2,\ 2$$

על המערך השני, נבחר את 12 כx:

$$8,\ 12,\ 12,\ 27,\ 91,\ 17$$

כעת צריך לעשות Quick-Sort על 4 המערכים הבאים:

$$-125$$
$$3,\ 3,\ 2,\ 2,$$
$$8$$
$$12,\ 27,\ 91,\ 17$$

את המערכים הראשון והשלישי אנחנו פשוט מחזירים.
את המערך השני: נקבל:

$$2,\ 2,\ 3,\ 3$$

נחלק לשני המערכים:

$$2,\ 2$$
$$3,\ 3$$

נחלק אותם שוב, הם ממויינים וכו'.
כך ממשיכים עד שכל מערך נהיה בגודל 1 ונקבל את המערך הממוין.
נסמן את זמן החישוב של Quick-Sort כ$Q\left(n\right)$ אזי:
שני השלבים הראשונים לוקחים פעולה 1.

השלב השלישי לוקח בערך n פעולות.
השלב הרביעי, הרקורסיה, לוקח בערך $2Q\left(\frac{n}{2}\right)$. לכן:

$$Q\left(n\right)=n+2Q\left(\frac{n}{2}\right)$$

למעשה, הנוסחה המדויקת יותר היא:

$$Q\left(n\right)=n+\frac{1}{n}\sum_{i=1}^{n}Q\left(i\right)+Q\left(n-i\right)$$

יש כמה שיטות לבדוק מה העלות של בעיית רקורסיה:

1. ניחוש + הוכחה באינדוקציה
2. הצבה
3. משפט Master

נבדוק את הבעיה הזו כרגע בשיטות:

1. תחילה, הצבה.

$$\begin{aligned}
Q\left(n\right) &= n+2Q\left(\frac{n}{2}\right)\\
&= n+2\left[\frac{n}{2}+2Q\left(\frac{n}{4}\right)\right]\\
&= n+n+4Q\left(\frac{n}{4}\right)\\
&= 2n+4\left[\frac{n}{4}+2Q\left(\frac{n}{8}\right)\right]\\
&= 2n+n+8Q\left(\frac{n}{8}\right)=...\\
&= k\cdot n+2^{k}Q\left(\frac{n}{2^{k}}\right)
\end{aligned}$$

בסוף נגיע למצב שבו $2^{k}=n$. נגיע למצב כזה כאשר $k=\log_{2}n$. ואז, נקבל:

$$Q\left(n\right)=\log_{2}n+n\cdot Q\left(1\right)=n\log_{2}n+c\cdot n=O\left(n\cdot\log_{2}n\right)$$

2. שיטת הניחוש. ננחש:

$$Q\left(n\right)=O\left(n\log_{2}n\right)$$

קיים c כך שהחל מn מסוים מתקיים:

$$Q\left(n\right)\leq c\cdot n\log_{2}n$$

נוכיח באינדוקציה.
נבדוק עבור $n=2$ (כי 1 לא עובד).

$$Q\left(2\right)=2+2Q\left(1\right)\leq c\cdot2\cdot1$$

עבור c מסוים.
נניח נכונות לכל $\frac{n}{2}\leq i<n$.

$$\begin{aligned}
Q\left(n\right) &= n+2Q\left(\frac{n}{2}\right)\\
&\leq n+2c\cdot\frac{n}{2}\log_{2}\frac{n}{2}\\
&= n+cn\cdot\left(\log_{2}n-1\right)\\
&= cn\log_{2}n+\left(n-cn\right)\\
&\leq cn\log_{2}n
\end{aligned}$$

עבור $c\geq1$.

נניח שמימשתי לא נכון את Quick Sory ויצא לי:

$$Q\left(n\right)=n^{2}+2Q\left(\frac{n}{2}\right)$$

נפתח בשיטת ההצבה:

$$\begin{aligned}
Q\left(n\right) & = & n^{2}+2Q\left(\frac{n}{2}\right)\\
 & = & n^{2}+2\left[\frac{n^{2}}{4}+2Q\left(\frac{n}{4}\right)\right]\\
 & = & n^{2}+2\left[\frac{n^{2}}{4}+2\left[\frac{n^{2}}{16}+Q\left(\frac{n}{8}\right)\right]\right]\\
 & = & n^{2}\left(1+\frac{1}{2}+\frac{1}{4}+...\right)+2^{k}Q\left(\frac{n}{2^{k}}\right)\\
 & = & 2n^{2}+n\cdot Q\left(1\right)=O\left(n^{2}\right)
\end{aligned}$$

לכן שימו לב שפה הרקורסיה לא שינתה את סדר הגודל של זמן הפעולה, כי n^{2} יותר משמעותי מ$\log_{2}n$.

משפט Master

בהינתן $f\left(n\right)=g\left(n\right)+bf\left(\frac{n}{a}\right)$.
יש שלוש אפשרויות:

1. אם קיים $\epsilon>0$ עבורו מתקיים:

$$g\left(n\right)=O\left(n^{\log_{a}\left(b\right)-\epsilon}\right)$$

אזי:

$$f\left(n\right)=\Theta\left(n^{\log_{a}b}\right)$$

2. אם קיים $\epsilon>0$ עבורו מתקיים:

$$g\left(n\right)=\Omega\left(n^{\log_{a}\left(b\right)+\epsilon}\right)$$

אזי

$$f\left(n\right)=\Theta\left(g\left(n\right)\right)$$

3. אם

$$g\left(n\right)=\Theta\left(n^{\log_{a}b}\right)$$

אזי

$$f\left(n\right)=\Theta\left(n^{\log_{a}b}\cdot\log n\right)$$

דוגמה

במקרה השני שלנו, בו:

$$Q\left(n\right)=n^{2}+2Q\left(\frac{n}{2}\right)$$

הנתונים הם:

$$\begin{aligned}
g\left(n\right) & = & n^{2}\\
a & =b= & 2
\end{aligned}$$

אזי

$$\log_a b = 1$$

כלומר

$$n^{\log_a b} = n$$

ולכן במקרה הזה $g(n)$ חסומה מלמעלה ע"י $n^{\log_a b}$ ולכן:

$$g(n) = \Omega\left(n^{\log_a b}\right)$$

ולפי המשפט:

$$Q(n) = \Theta(g(n)) = \Theta\left(n^2\right)$$

במקרה הקודם, של Quick-Sort אמיתי, היה:

$$\begin{aligned} g(n) &= n \\ n^{\log_a b} &= n \end{aligned}$$

לכן במצב הזה:

$$g(n) = \Theta\left(n^{\log_a b}\right)$$

ולכן

$$Q(n) = \Theta(n \log n)$$

דוגמה למקרה הראשון

למשל:

$$Q(n) = 1 + 2Q\left(\frac{n}{2}\right)$$

בשיטת ההצבה, נקבל:

$$\begin{aligned} Q(n) &= 1 + 2\left[1 + 2Q\left(\frac{n}{4}\right)\right] \\ &= 1 + 2\left[1 + 2\left[1 + Q\left(\frac{n}{8}\right)\right]\right] \\ &= .. = 1 + 2 + 4 + ... + 2^k + 2^k Q\left(\frac{n}{2^k}\right) \\ &= n + n \cdot Q(1) = O(n) \end{aligned}$$

בשיעור הבא נסתכל על זה לפי משפט Master.