מבנה נתונים ואלגוריתמים - תרגול 2

6 בנובמבר 2011

נוסחאות עם רקורסיה

לאורך הסמסטר ניתקל בנוסחאות כמו:

$$T\left(n\right)=\begin{cases} 2T\left(\frac{n}{2}\right)+n & n>1\\ 1 & n=1 \end{cases}$$

נרצה למצוא פונק' מפורשת $f\left(n\right)$ כך שמתקיים $T\left(n\right)=\Theta\left(f\left(n\right)\right)$.

דרכים למצוא את $f\left(n\right)$

דרך א' - ניחוש

מנחשים $f\left(n\right)$ ומוכיחים באינדוקציה.

דוגמה

$$T\left(n\right)=\begin{cases} T\left(n-1\right)+n^{2} & n>1\\ 1 & n=1 \end{cases}$$

ננחש: $T\left(n\right)=n^{3}$ וננסה למצוא $c_{1},\,c_{2}$ כך שמתקיים:

$$c_{1}n^{3}\leq T\left(n\right)\leq c_{2}n^{3}$$

אם $c_{1}<1$ ו$c_{2}>1$ אז:

$$c_{1}\cdot1\leq T\left(1\right)\leq c_{2}\cdot1$$

ננסה לבחור c_{2} כך ש$T\left(n\right)\leq c_{2}n^{3}$.
נניח שזה נכון לכל $n^{'}<n$.

$$\begin{aligned} T\left(n\right) & = T\left(n-1\right)+n^{2}\leq c_{2}\left(n-1\right)^{3}+n^{2}\\ & = c_{2}n^{3}-3c_{2}n^{3}+3c_{2}n-c_{2}+n^{2}\\ & = c_{2}n^{3}-\left[\left(3c_{2}-1\right)n^{2}-3c_{2}n+c_{2}\right] \end{aligned}$$

אנו רוצים שלכל n יתקיים:

$$\left(3c_{2}-1\right)n^{2}-3c_{2}n+c_{2}\geq0$$

נבחר $c_{2}=3$ ואז אי השוויון הוא

$$8n^{2}-9n+3\geq0$$

אין שורשים כי

$$9^{2}-4\cdot8\cdot3=-15<0$$

ולכן אין שורשים לביטוי ולכן אי השוויון נכון תמיד.
לכן הוכחנו באינדוקציה שמתקיים עבור $c_{2}=3$:

$$T\left(n\right)\leq c_{2}n^{3}$$

דרך ב' - פיתוח הנוסחה

פותחים את הרקורסיה ומוצאים נוסחה מפורשת.

דוגמה

$$T(n)=\begin{cases}\sqrt{n}T(\sqrt{n})+n & n\geq 2\\ 1 & n\leq 2\end{cases}$$

אזי:

$$\begin{aligned}T(n) &= \sqrt{n}T\left(\sqrt{n}\right)+n\\ &= n^{\frac{1}{2}}T\left(n^{\frac{1}{2}}\right)+n\\ &= n^{\frac{1}{2}}\left(n^{\frac{1}{4}}T\left(n^{\frac{1}{4}}\right)+n^{\frac{1}{2}}\right)+n\\ &= n^{1-\frac{1}{4}}T\left(n^{\frac{1}{4}}\right)+n+n\\ &= n^{1-\frac{1}{4}}\left(n^{\frac{1}{8}}T\left(n^{\frac{1}{8}}\right)+n^{\frac{1}{4}}\right)+2n\\ &= n^{1-\frac{1}{8}}T\left(n^{\frac{1}{8}}\right)+3n\\ &= n^{1-\frac{1}{2^k}}T\left(n^{\frac{1}{2^k}}\right)+kn\end{aligned}$$

כעת נבדוק מתי $n^{\frac{1}{2^k}}\leq 2$:

$$\begin{aligned}\lg n^{\frac{1}{2^k}} &\leq \lg 2\\ \frac{1}{2^k}\lg n &\leq 1\\ \lg n &\leq 2^k\\ \lg\lg n &\leq k\end{aligned}$$

(הערה: כאשר נכתוב $\lg$ אנו מתכוונים ל-$\log_2$, וכאשר נכתוב $\log$ אנו מתכוונים ל-$\ln$).
ניקח $k=\lg\lg n$ ונקבל מהפיתוח הקודם:

$$\begin{aligned}T(n) &= \frac{n}{n^{\frac{1}{2^k}}}T\left(n^{\frac{1}{2^k}}\right)+n\lg\lg n\\ &= \frac{n}{2}T(2)+n\lg\lg n\\ &= \frac{n}{2}+n\lg\lg n=\Theta(n\lg\lg n)\end{aligned}$$

דרך ג' - משפט המאסטר

אם $0<a,\, b\in\mathbb{R}$ ונתון:

$$T(n)=\begin{cases}aT\left(\frac{n}{b}\right)+f(n) & n>n_0\\ g(n) & n\leq n_0\end{cases}$$

אזי:

$$T(n)=\begin{cases}\Theta\left(n^{\log_a b}\right) & \exists\epsilon>0:\ f(n)=O\left(n^{\log_a b-\epsilon}\right)\\ \Theta\left(n^{\log_a b}\log n\right) & f(n)=\Theta\left(n^{\log_a b}\right)\\ \Theta(f(n)) & \begin{array}{c}\exists\epsilon>0:\ f(n)=\Omega\left(n^{\log_a b+\epsilon}\right)\\ \exists c<1,\, n_0:\ \forall n>n_0\ bf\left(\frac{n}{b}\right)\leq c\cdot f(n)\end{array}\end{cases}$$

(החלק השני של המקרה האחרון בדר"כ מתקיים, בהרצאה לוזון לא כתב אותו אז כנראה לא ממש צריך אותו והוא לא מתקיים רק במקרים מיוחדים).

דוגמאות

.1

$$T\left(n\right)=7T\left(\frac{n}{2}\right)+n^{2}$$

במקרה הזה

$$b=7,\,a=2$$

אזי

$$\log_{a}b=\log_{2}7>\log_{2}4=2$$

ומתקיים

$$n^{2}=O\left(n^{\log_{a}b-\epsilon}\right)$$

לכן

$$T\left(n\right)=\Theta\left(n^{\log_{2}7}\right)=\Theta\left(n^{2.81}\right)$$

.2

$$T\left(n\right)=4T\left(\frac{n}{2}\right)+\Theta\left(n^{2}\right)$$

במקרה זה

$$a=2,\,b=4$$

אזי

$$\log_{a}b=\log_{2}4=2$$

ומתקיים

$$\Theta\left(n^{2}\right)=\Theta\left(n^{\log_{a}b}\right)$$

לכן

$$T\left(n\right)=\Theta\left(n^{\log_{a}b}\log n\right)=\Theta\left(n^{2}\log n\right)$$

הערה: בתוך $O,\,\Theta,\,\Omega$ בדר"כ אין משמעות לבסיס הלוגריתם, כי $\log_{r}n=\frac{\ln n}{\ln r}=\Theta\left(\ln n\right)$.

.3

$$T\left(n\right)=4T\left(\frac{n}{5}\right)+\Theta\left(n\right)$$

במקרה זה

$$\log_{5}4<\log_{5}5=1$$

לכן קיים $\epsilon>0$ כך ש:

$$n=\Omega\left(n^{\log_{5}4+\epsilon}\right)$$

ומתקיים המקרה השלישי, לכן

$$T\left(n\right)=\Theta\left(n\right)$$

.4

$$T(n) = 4T\left(\frac{n}{5}\right) + \Theta(n \lg n)$$

שוב,

$$\log_5 4 < \log_5 5 = 1$$

לכן קיים $\epsilon > 0$ כך ש

$$n \lg n = \Omega\left(n^{\log_5 4+\epsilon}\right)$$

ולכן

$$T(n) = \Theta(n \lg n)$$

.5

$$T(n) = 5T\left(\frac{n}{5}\right) + n \log_5 n$$

במקרה זה

$$\log_5 5 = 1$$

אבל

$$f(n) \neq \Theta(n)$$

ולא קיים $\epsilon > 0$ כך ש

$$f(n) = \Omega\left(n^{1+\epsilon}\right)$$

או

$$f(n) = O\left(n^{1-\epsilon}\right)$$

לכן אי אפשר להשתמש במשפט המאסטר.
בבית - פתרו עם פיתוח וגלו שמתקיים:

$$T(n) = \Theta\left(n \log^2 n\right)$$

הערה

משפט המאסטר תקף גם (במיוחד) לרקורסיות עם עיגול כלפי מעלה או כלפי מטה.
לדוגמה, את תרגיל 3 היה ניתן להחליף ב:

$$T(n) = nT\left(\left\lceil\frac{n}{5}\right\rceil\right) + 2T\left(\left\lfloor\frac{n}{5}\right\rfloor\right) + n$$

והפתרון היה תקף.

אלגוריתמים

אלגוריתם = סדרת הוראות למכונה דמיונית (בדר"כ מחשב). ההוראות הן פעולות בסיסיות של המכונה.
סיבוכיות הזמן של האלגוריתם = כמה זמן לוקח לאלגוריתם לרוץ על המכונה.
זה תלוי בכמה זמן לוקחת כל פעולה בסיסית, ובכל מקרה זה Θ של מספר הפעולות הבסיסיות.
אם למכונה שלנו יש תאי זיכרון, סיבוכיות הזיכרון של האלגוריתם היא Θ של כמות התאים שהוא צריך כדי לרוץ.

הערה

אם הקלט לאלגוריתם הוא רשימה, בדר"כ סיבוכיות הזמן והזיכרון הן פונק' של אורך הרשימה.
אם הקלט הוא מספר, אז בדר"כ הסיבוכיות תהיה פונק' של המספר.

פעולות בסיסיות (שבדר"כ) מותרות

"כל הפעולות בשפת c":

- פעולות על מספרים - $+, -, \div, \cdot, \hat{}, \&, <, >, \mathrm{mod}$
- השוואה בין משתנים בסיסיים
- הגדרת משתנים חדשים
- הקצאת מערך
- לולאות $(\mathrm{for,\ while})$
- פיצולים $(\mathrm{if,\ else})$
- רקורסיה, קריאה לפונ' עזר.

דוגמאות

1. קלט - מספר n.

```
t = n
i = 2
while t¿1
 if t%i=0
  print i
  t=t/i
 else
  i+=1
 endif
endwhile
```

האלגוריתם מדפיס את הפירוק לגורמים של n.
האלגוריתם לא יעשה יותר מ-$\sqrt{n}$ מעברים על לולאת הwhile.
מצד שני אם n ראשוני הוא עושה $\lfloor\sqrt{n}\rfloor$ מעברים על הלולאה.
לכן סיבוכיות הזמן היא $O\left(\sqrt{n}\right)$ (במקרה הגרוע ביותר).
במקרה הטוב ביותר, כאשר $n = 2^k$ הלולאה תתבצע $\lg n$ פעמים.