מבני נתונים ואלגוריתמים - תרגול 13

22 בינואר 2012

תרגיל

נאמר שמחרוזת X מחזורית אם $\exists n > 1$ ומחרוזת Y כך ש$Y^n = X$.
כיצד תבדקו האם מחרוזת X היא מחזורית ב$O\left(\mathrm{len}\left(X\right)\right)$?

פתרון

אם X מחזורית, אז X תת מחרוזת שאינה רישא או סיפא של XX.

טענה

X מחזורית $\iff$ X תת מחרוזת של XX שאינה רישא או סיפא.
אם הוכחנו את הטענה, אז האלגוריתם הבא בודק אם X מחזורית:

אלגוריתם 1 פתרון תרגיל - בדיקת מחזוריות

---

1. בנה את $S = XX$
2. חפש את X בתוך S (בעזרת KMP)
3. אם X מופיעה בS במקום שונה מ-0 ומ$\mathrm{len}\left(X\right)$ אז X מחזורית. אחרת, לא.

הסיבוכיות היא $O\left(\mathrm{len}\left(XX\right)\right) = O\left(\mathrm{len}\left(X\right)\right)$.

הוכחת הטענה

נכתוב $XX = AXB$ כאשר A, B לא ריקות.
$A = X_{\mathrm{len}(A)}$ ובאופן דומה $B = {}_{\mathrm{len}(A)} X$.
אזי $X = AB$.
נציב ב$XX = AXB$ ונקבל:

$$ABAB = AABB$$

נצמצם ונקבל

$$BA = AB$$

לכן $BA = AB$.
סיימנו הודות ללמה הבאה:

למה

אם קיימות מחרוזות לא ריקות A, B כך ש$X = AB = BA$ אז X מחזורית.
מספיק להוכיח שקיימים $n, m \geq 1$ ומחרוזת Y כך ש$A = Y^n$ ו$B = Y^m$.
נוכיח באינדוקציה על $\mathrm{len}\left(X\right)$.
אם $\mathrm{len}\left(A\right) = \mathrm{len}\left(B\right)$:

$$A = X_{\mathrm{len}(A)} = X_{\mathrm{len}(B)} = B$$

נבחר $n=m=1$ ו$Y=A=B$.
אם $\operatorname{len}(A)\neq\operatorname{len}(B)$:
בה"כ נניח ש$\operatorname{len}(A)>\operatorname{len}(B)$.

$$X=AB=BA$$

בהכרח B רישא אמיתית של A, אז קיימת C לא ריקה כך ש$A=BC$.
לכן,

$$\begin{aligned} X &= AB=BCB \\ X &= BA=BBC \end{aligned}$$

נצמצם B ונקבל

$$BC=CB$$

לפי אינד' קיימים $n,m\geq 1$ וY כך ש$B=Y^m$ ו$C=Y^n$ לכן $A=BC=Y^{n+m}$ וסיימנו.

BM - המשך

T מחרוזת, P תבנית, $m=\operatorname{len}(P)$, $i=m-1$, $q=m-1$.

אלגוריתם 2 אלגוריתם BM

```
כל עוד i < len(T):
}
  כל עוד T(i) = P[q] ו q > 0:
  }
    i-=1
    q-=1
  {
  אם q = -1
    מצאנו (באינדקס i)
  אחרת:
  }
    q = m-1
    i+= max{GST[q] + BST[T(i)]}
  {
{
```

הטבלאות - עבור התבנית $P=aabcba$:

BST	אות
4	a
1	b
2	c
6	אחרת

GST	j
1	5
5	4
7	3
8	2
9	1
10	0

תכנון דינמי

הרעיון

יש בעיה שאפשר לפתור ברקורסיה, אבל הרבה תתי מקרים של הבעיה מתלכדים, אז במקום לבצע רקורסיה פותרים קודם את כל תתי המקרים (תוך הימנעות מפתירה חוזרת של אותו תת מקרה) ואז

פותרים את הבעיה.

דוגמה

סדרת פיבונאצ'י - $F_{n+2} = F_{n+1} + F_n$.
ברקורסיה, נצטרך לחשב את F_n ברקורסיה עבור F_{n+2} ולחשב אותו שוב ברקורסיה עבור F_{n+1}, ואז מחשבים אותו פעמיים.

פתרון בתכנון דינמי

נחשב את $F_0, F_1, F_2, F_3 ...$ לפי הסדר.
$F_0 = F_1 = 1$.
עבור $2 \leq i \leq n$:
$F_i = F_{i-1} + F_{i-2}$.
החזר את F_n.
סיבוכיות - $O(n)$.
זיכרון - מספיק לשמור רק את F_i, F_{i-1}, F_{i-2} בכל צעד בלולאה. במימוש חכם, הזיכרון יהיה $O(1)$.

תרגיל

נתונה סדרת מספרים $\{x_i\}_{i=1}^{n}$.
רוצים למצוא את אורך תת הסדרה הארוכה ביותר שהיא מונו' עולה.

פתרון

נגדיר $\ell(i)$ להיות אורך תת הסדרה המונוטונית עולה המקסימלית של $x_1, ..., x_i$.

טענה

$$\ell(i) = 1 + \max_{\substack{j \leq i \\ x_j \leq x_i}} \ell(j)$$

הוכחה

תהי $x_{n_1}, x_{n_2}, ..., x_{n_k}$ תת הסדרה המונו' עולה הארוכה ביותר של $x_1, ..., x_i$ כך ש$i = n_k$.
אזי $x_{n_1}, x_{n_2}, ..., x_{n_{k-1}}$ תת סדרה מונו' מקס' המסתיימת ב$x_{n_{k-1}}$ (כי אחרת הייתה סדרה ארוכה יותר $x_{m_1}, ..., x_{m_r=n_{k-1}}$ ואז היה מתקיים ש$x_{m_1}, ..., x_{m_r}, x_{n_k}$ ארוכה יותר מאורך k המסתיימת בx_{n_k}).
לכן, $\ell(n_k - 1) + 1 = \ell(i)$, לכן:

$$\ell(i) \leq 1 + \max_{\substack{j < i \\ x_j \leq x_i}} \ell(j)$$

(כי ממקסמים גם על $\ell(n_{k-1})$).
הכיוון השני ($\geq$) מתקיים כי $\ell(i) \geq 1 + \ell(j)$ עבור כל j המקיים $x_i \geq x_j$.

אלגוריתם 3 פתרון התרגיל

$\ell(0) = 0$
עבור $i = 1, .., n$:
}
$max = 0$
עבור $j = 1, ..., i-1$
אם $x_j < x_i$:
$max = \max\{\ell(j) + 1, max\}$
{
החזר את $\max\{\ell(i)\}$ עבור $i = 1, ..., n$.

הסיבוכיות:
הלולאה הראשונה מתבצעת n פעמים, הלולאה השניה מתקיימת i פעמים באיטרציה הi, לכן מתבצעות $O\left(n^2\right)$ פעולות.
זיכרון:
$O\left(n\right)$.

תרגיל (כפל מטריצות)

עובדה: לכפול מטריצת $n\times k$ במטריצת $k\times m$ לוקח nkm עבודה.
נתונות מטריצות $A_1,\,A_2,\,..,\,A_n$ עם מימדים $A_i\in\mathbb{R}^{n_{i-1}\times n_i}$.
רוצים לחשב את המכפלה:

$$A_1A_2...A_n$$

באופן היעיל ביותר.

פתרון

נגדיר $T_{m,\,k}$ הזמן המינימלי הדרוש לכפול את $A_m,\,...,\,A_k$.
אנחנו רוצים את $T_{1,\,n}$.
מתקיים:

$$\begin{aligned} T_{m,\,k} &= \min_{m\le r<k}\left\{T_{m,\,r}+T_{r+1,\,k}+n_{m-1}n_rn_k\right\} \\ T_{k,\,k} &= 0 \end{aligned}$$

נחשב את $T_{1,\,1},\,T_{2,\,2},\,...,\,T_{n,\,n}$, מהם נחשב את $T_{1,\,2},\,T_{2,\,3},\,...,\,T_{n-1,\,n}$, וכו'.

אלגוריתם 4 פתרון התרגיל

לכל $1\le i\le n$:
$T_{i,\,i}=0$
עבור $k=1,\,...,\,n-1$:
}
עבור $i=1,\,...,\,n-k$:
}
$T_{i,\,i+k}=\min_{i\le r<i+k}\left\{T_{i,\,r}+T_{r+1,\,i+k}+n_{i-1}n_rn_{i+k}\right\}$
{
{
החזר את $T_{1,\,n}$.

סיבוכיות:
הלולאה הראשונה רצה n פעמים, השניה $n-k$ באיטרציה הk, ומציאת המינימום לוקחת $O\left(k\right)$ פעולות בכל איטרציה, לכן סה"כ הסיבוכיות היא $O\left(n^3\right)$.
זיכרון:
שומרים את $T_{i,\,j}$ לכל $1\le i\le j\le n$ כלומר $O\left(n^2\right)$ זיכרון.