פתרונות לממ"ן 15 בקורס "אלגוריתמים", סמסטר ב' 2006
(הוכן בשיתוף עם יואב גיורא)

שימו לב: לעיתים חסרים פרטים בפתרון ואתם מתבקשים לוודא שהינכם יודעים להשלים את הפרטים החסרים. בכל מקרה כזה, הדבר מצויין במפורש, ואין לראות בפתרון הנתון תשובה מלאה.

שאלה 1

תזכורת: מטריצה משולשית תחתונה היא מטריצה שכל איבריה מעל האלכסון הראשי שווים ל-0.

נניח שעלות העלאה בחזקה שלישית של מטריצה משולשית תחתונה בגודל $n\times n$ היא $s(n)$. הראו כי בהינתן שתי מטריצות כלשהן A ו-B בגודל $n\times n$ ניתן לחשב את מכפלתן $A\cdot B$ בזמן $O(s(3n))$ (כלומר, העלאה בחזקה שלישית של מטריצות משולשיות תחתונות היא קשה לפחות כמו מכפלת מטריצות כלליות).

תשובה

נראה כי ניתן להכפיל שתי מטריצות בגודל $n\times n$ על ידי רדוקציה לבעיית העלאה בחזקה שלישית של מטריצה משולשית תחתונה מגודל $3n\times 3n$. בהינתן A ו-B מטריצות כלשהן מגודל $n\times n$ נבנה את המטריצה המשולשית התחתונה הבאה מגודל $3n\times 3n$:

$$\begin{pmatrix} 0 & 0 & 0 \\ B & I & 0 \\ 0 & A & 0 \end{pmatrix}$$

עלות בנייתה היא $O((3n)^2)$. נעלה אותה בחזקה שלישית, בעלות $O(s(3n))$. התוצאה היא המטריצה הבאה:

$$\begin{pmatrix} 0 & 0 & 0 \\ B & I & 0 \\ AB & A & 0 \end{pmatrix}$$

ולכן ע"י שליפת תת-המטריצה המתקבלת מ-n^2 הכניסות התחתונות משמאל, תהיה בידינו המכפלה הדרושה.

ברור כי $s(n)$ הוא לפחות n^2, כי רק כתיבת התוצאה של העלאה בחזקה שלישית דורשת $O(n^2)$. לכן הסיבוכיות הכוללת היא $O(s(3n))$.

שאלה 2

גירסה של האלגוריתם של שטראסן, מעט שונה מזו שבספר הלימוד, מבוססת על הזהויות הבאות:

כדי לחשב $A\cdot B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} e & g \\ f & h \end{pmatrix} = \begin{pmatrix} r & s \\ t & u \end{pmatrix}$ מחשבים קודם:

$s_1 = c + d$	$m_1 = s_2 \cdot s_6$	$t_1 = m_1 + m_2$
$s_2 = s_1 - a$	$m_2 = a \cdot e$	$t_2 = t_1 + m_4$
$s_3 = a - c$	$m_3 = b \cdot f$	
$s_4 = b - s_2$	$m_4 = s_3 \cdot s_7$	$r = m_2 + m_3$
$s_5 = g - e$	$m_5 = s_1 \cdot s_5$	$s = t_1 + m_5 + m_6$
$s_6 = h - s_5$	$m_6 = s_4 \cdot h$	$t = t_2 - m_7$
$s_7 = h - g$	$m_7 = d \cdot s_8$	$u = t_2 + m_5$
$s_8 = s_6 - f$		

הוכיחו את נכונות האלגוריתם והשוו את יעילותו ליעילות האלגוריתם של שטראסן, בגירסתו המוכרת לכם מספר הלימוד.

תשובה

נראה כי המשוואות המתקבלות אכן נכונות.

$$A \cdot B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} e & g \\ f & h \end{pmatrix} = \begin{pmatrix} ae+bf & ag+bh \\ ce+df & cg+dh \end{pmatrix} = \begin{pmatrix} r & s \\ t & u \end{pmatrix}$$

$r = m_2 + m_3 = a \cdot e + b \cdot f$

$s = t_1 + m_5 + m_6 = m_1 + m_2 + m_5 + m_6 = s_2 \cdot s_6 + a \cdot e + s_1 \cdot s_5 + s_4 \cdot h + = (s_1 - a)(h - s_5) + a \cdot e + (c + d)(g - e)$
$+ (b - s_2) \cdot h = (c + d - a)(h - g + e) + a \cdot e + (c + d)(g - e) + (b - s_1 + a) \cdot h = (c + d - a)(h - g + e) + a \cdot e$
$+ (c + d)(g - e) + (b - c - d + a) \cdot h = \ldots = a \cdot g + b \cdot h$

$t = t_2 - m_7 = t_1 + m_4 - m_7 = m_1 + m_2 + m_4 - m_7 = \ldots = c \cdot e + d \cdot f$

$u = t_2 + m_5 = t_1 + m_4 + m_5 = m_1 + m_2 + m_4 + m_5 = \ldots = c \cdot g + d \cdot h$

יש 15 פעולות חיבור ו-7 פעולות כפל, לעומת האלגוריתם המקורי של שטראסן שבו 18 פעולות חיבור ו-7 פעולות כפל. אסימפטוטית זו אותה סיבוכיות – $O(n^{\log_2 7})$ – אבל הקבוע מעט קטן יותר.

שאלה 3

נניח כי נתון אלגוריתם A המחשב מכפלת שתי מטריצות שגודלן 4×4 המורכבות מאיברים מעל חוג כלשהו R (ראו תזכורת בסוף השאלה) ומשתמש ב-x מכפלות של איברים מ-R.

א. תארו אלגוריתם יעיל המשתמש ב-A כקופסה שחורה, ומכפיל שתי מטריצות בגודל $n \times n$ של מספרים שלמים (הניחו ש-n חזקה של 4).
הוכיחו את נכונות האלגוריתם ונתחו את סיבוכיותו (כפונקציה של n ו-x).
הניחו שפעולות אריתמטיות על שלמים מתבצעות בסיבוכיות $O(1)$.

ב. מה צריך להיות הערך של x כדי שאלגוריתם זה יהיה טוב יותר מהאלגוריתם של שטראסן?

תזכורת ורמז: חוג הוא אוסף של איברים שמוגדרות עליהם פעולת חיבור ופעולת כפל, המקיימות תכונות מסוימות (ביניהן אסוציאטיביות, דיסטריבוטיביות ועוד).
דוגמאות לחוגים: המספרים הממשיים עם פעולות הכפל והחיבור הרגילות, המספרים השלמים עם פעולות הכפל והחיבור הרגילות, המטריצות בגודל $n \times n$ מעל המספרים השלמים עם פעולות חיבור מטריצות וכפל מטריצות.

תשובה

א. בהינתן שתי מטריצות B_1 ו-B_2 בגודל $n \times n$ נחלק כל מטריצה ל-16 מטריצות בגודל $n/4 \times n/4$. למעשה, כעת המטריצה B_1 היא מטריצה 4×4 שאיבריה הן מטריצות בגודל $n/4 \times n/4$ וכך גם B_2.
נשתמש באלגוריתם A כדי להכפיל את B_1 ו-B_2. האלגוריתם A משתמש ב-x מכפלות של מטריצות בגודל $n/4 \times n/4$ ועוד $O(1)$ פעולות של חיבור מטריצות בגודל $n/4 \times n/4$. נמשיך מכאן רקורסיבית (כלומר, נבצע כל מכפלה של מטריצות בגודל $n/4 \times n/4$ על ידי חלוקת כל אחת מהן ל-16 מטריצות ושימוש ב-A).
הוכחת הנכונות ברורה (וודאו כי הנכם יודעים לכתוב אותה!).
ניתוח הסיבוכיות: $T(n) = x \cdot T(n/4) + O(n^2) = O(n^{\log_4 x})$ (הגורם הריבועי הוא מפעולות החיבור).

ב. עלות האלגוריתם של שטראסן היא $O(n^{\log_2 7})$. מתקיים $\log_4 x < \log_2 7 = \log_4 49$ כאשר $x < 49$.

שאלה 4

חשבו את הביטויים הבאים :

א. $DFT_m\left(x^n\right)$ כאשר $m \le n$ וכן m מחלק את n.

ב. $DFT_{n+1}\left(\sum_{i=0}^{n} x^i\right)$.

תשובה

הערה: בשאלה זו DFT מנוסח על פולינום. ע"פ ההגדרות בחומר הלימוד, הדרך המדויקת לכתוב זאת היא במונחים של וקטור מקדמים. כלומר, עבור סעיף א' יש לחשב DFT מסדר m על וקטור המקדמים באורך $n+1$ (0, 0, ..., 0, 1) ועבור סעיף ב' יש לחשב DFT מסדר $n+1$ על וקטור המקדמים (1, 1, ..., 1).

א. DFT צריך להחזיר את ערכי הפולינום ב-m שורשי היחידה מסדר m. התוצאה תהיה וקטור באורך m של 1.

הוכחה: $(k' = n/m \text{ integer}) \quad x^n(w_m^k) = (e^{i\frac{2\pi k}{m}})^n = e^{i\cdot 2\pi k\cdot k'} = 1 \quad 0 \le k \le m-1$

$\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$. עבור $x \neq 1$: $\frac{1-x^{n+1}}{1-x}(w_{n+1}^k) = 0$. עבור $x=1$ התוצאה היא $n+1$. ולכן סה"כ הוקטור המבוקש הוא (n+1, 0, 0, ..., 0).

שאלה 5

בהינתן טקסט $T = t_0, t_1, \ldots, t_{n-1}$ באורך n, ותבנית $P = p_0, p_1, \ldots, p_{m-1}$ באורך m, מא"ב $\{a, b\}$, תארו אלגוריתם יעיל המוצא לכל אינדקס $0 \le j \le n-m$ את מספר האי-התאמות בין התבנית P לבין המחרוזת $t_j, t_{j+1}, \ldots, t_{m+j-1}$.

למשל, אם התבנית P היא *aabba* והטקסט T הוא *ababab*, אז האלגוריתם צריך לתת את הפלט הבא:

אינדקס 0: 2

אינדקס 1: 3

אם T הוא *bbbbbbb* ו-P היא *aabba*, האלגוריתם צריך לתת את הפלט הבא:

אינדקס 0: 3

אינדקס 1: 3

אינדקס 2: 3

רמז: התאימו את a ל- 1 ואת b ל- -1.

הוכיחו את נכונות האלגוריתם ונתחו את סיבוכיותו.

בונוס: בהינתן טקסט T באורך n ותבנית P באורך m, בא"ב בן k אותיות, תארו אלגוריתם יעיל, המוצא את כל האינדקסים $0 \le j \le n-m$ כך ש:

$$p_0 \cdots p_{m-1} = t_j \cdots t_{m+j-1}$$

תשובה

תחילה נציג פתרון בסיבוכיות $O(n \log n)$. בהמשך לרמז, נחליף את a ב-1 ואת b ב- (-1), ונבנה מהטקסט T פולינום $t(x)$ שמקדמיו נקבעים על ידי תווי המחרוזת T, כלומר $t(x) = t_{n-1}x^{n-1} + \ldots + t_1 x + t_0$. באופן דומה, נבנה מהתבנית Reverse(P), שהיא המחרוזת P בהיפוך סדר התווים (כלומר, "נקרא" את P מהסוף להתחלה), פולינום $p(x) = p_0 x^{m-1} + \ldots + p_{m-2}x + p_{m-1}$.

כעת נחשב את פולינום המכפלה $r(x)=t(x)\cdot p(x)$ בסיבוכיות $O(n\log n)$ (על ידי DFT). דרגת פולינום המכפלה תהיה $(n-1)+(m-1)=n+m-2$ ונסמן אותו $r(x)=r_{n+m-2}x^{n+m-2}+...+r_1x+r_0$. אנו מעוניינים במקדמים r_j רק עבור $m-1\le j\le n-1$ כיוון שהם מוגדרים על ידי $r_j=p_0t_{j-(m-1)}+...+p_{m-2}t_{j-1}+p_{m-1}t_j$. סכום זה קשור למספר ההתאמות בין התבנית $p_0,...,p_{m-1}$ לקטע הטקסט $t_{j-(m-1)},...,t_j$ כדלקמן: כל התאמה ביניהם תורמת +1 לסכום, וכל אי-התאמה תורמת (-1) לסכום.

קיבלנו כי המקדם r_j הוא ההפרש בין מספר ההתאמות למספר האי-התאמות. נזכור כי הסכום של מספר ההתאמות ושל מספר האי-התאמות הוא בדיוק m, ולכן נוכל לחשב את מספר האי-התאמות, שנסמן אותו על ידי q_j, כדלקמן: ברור שמספר ההתאמות הוא $m-q_j$, ולכן מתקיים כי $r_j=1\cdot(m-q_j)+(-1)\cdot q_j=m-2q_j$. אם כן, מספר האי-התאמות הוא $q_j=(m-r_j)/2$.

ניתן לשפר את הפתרון ולקבל סיבוכיות $O(n\log m)$. נחתוך את הטקסט T לתת-מחרוזות באורך m^2 ונבצע עבור כל אחת מהן את האלגוריתם שתואר לעיל. נשים לב כי בנקודת "ההדבקה" בין המחרוזות יש תחום שלא נבדק (כיון שחלקו שייך למחרוזת אחת וחלקו למחרוזת אחרת), ולכן נוסיף לכל מחרוזת תחילית באורך m-1 השייכת לסוף המחרוזת הקודמת (כלומר, בין המחרוזות תהיה "חפיפה" באורך m-1). סיבוכיות החיתוך של T היא ליניארית. כיוון שמספר התת-מחרוזות הוא n/m^2 ואורך כל אחת הוא m^2+m, בסך הכל סיבוכיות האלגוריתם היא $O\left(\frac{n}{m^2}\cdot(m^2+m)\log(m^2+m)\right)=O(n\log m)$.

<u>בונוס</u>: נקודד את הטקסט ואת התבנית בצורה בינארית. לקידוד כל אות יידרשו $\lceil\log k\rceil$ סיביות. כעת נפעיל את האלגוריתם שתואר לעיל עבור המחרוזות. נשים לב כי בקידוד זה יש התאמה בין התבנית לקטע טקסט אם ורק אם יש התאמה כזו גם במקור. (נעיר כי מספר האי-התאמות בקידוד אינו מאפשר לקבוע באופן יחיד את מספר האי-התאמות במקור). סיבוכיות האלגוריתם היא $O(n\log k\cdot\log(m\cdot\log k))=O(n\log k(\log m+\log\log k))$.
נשים לב כי אם $k>m$ אזי ישנם תווים שלא מופיעים בתבנית וניתן לפסול מייד מחרוזות שמכילות תווים אלו. נעשה זאת ביעילות על ידי החלפת כל התווים האלו בתו אחד שאינו נמצא בתבנית וכך נקבל כי $k=m+1$. מכאן נסיק כי log k < m+1 וסיבוכיות האלגוריתם $O(n\cdot\log k\cdot\log m)$.