5.12.16 בס"ד

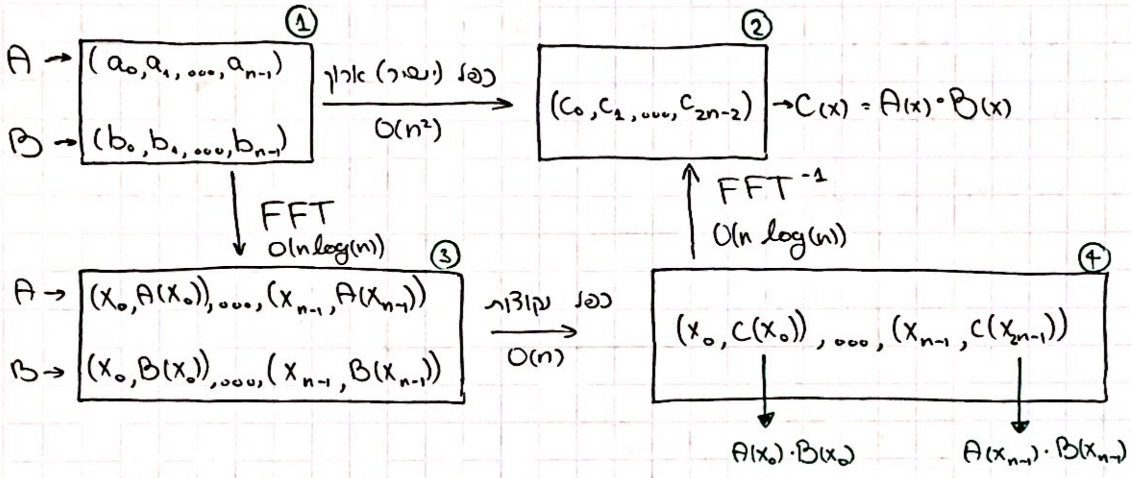
אלגוריתמים - תרגול 4

תזכורת: בהרצאה למדנו את האלגוריתם FFT, שמוצא מהר לנו מכפלה של פולינומים מדרגה n.

ראינו שכפל ישיר של 2 הפולינומים לוקח $O(n^2)$.

$$A(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$$

$$B(x) = b_0 + b_1x + \dots + b_{n-1}x^{n-1}$$



← ייצוג פולינום ע"י נקודות אומנם יותר מהיר לבצע מכפלת פולינומים אך ייצוג ע"י מקדמים ניתן למצוא את הערך של הפונק' בנק' מסויימת בצורה יחסית ישירה.

היינו רוצים להעביר את הייצוג ב-1 לייצוג ב-3 ורק לאחר מכן להכפיל ולהגיע ל-4 אך המעבר בין 1 ל-3 ייקח לנו $O(n)$ לכל נקודה כפול n נקודות סה"כ $O(n^2)$ וכך לא חסכנו כלום.

פתרון - נעזר בעזרת FFT וכך המעבר ייקח לנו $O(n\log(n))$ בכך שלא נבחר סתם נקודות אקראיות אלא נבחר את שורשי היחידה. (ω_n)

← בשיטה זו אנחנו צריכים לרפד את A ו-B באפסים עד a_{2n} ו-b_{2n} וכך יהיו לנו מספיק נקודות ב-C כדי למצוא את הפולינום מדרגה $2n-1$.

העברה של C מייצוג של נקודות (מ-4 ל-2) לא בעזרת מכפלת מטריצות שלוקחת במקרה הסדיר $O(n^3)$ או $O(n^2)$, שוב לא התקדמנו.

לכן נכניס את הנקודות ל-FFT^{-1} וכך נקבל את המעבר הזה שוב ב-$O(n\log(n))$

מהו ה- FFT הזה?

בהינתן וקטור באורך n הוא מוצא את הערכים בשורשי היחידה של הפולינום.

(הערכים של הפולינום בשורשי היחידה מהסדר של אורך הוקטור (n) נקראים DFT (וקטור))

(*) נניח ש-n הוא חזקה שלמה של 2 (אם לא נרפד באפסים עד החזקה של 2 הכי קרובה וזה לא משנה את זמן הריצה כי זה מקסימום מכפיל ל-$2n$.

$A = (a_0, a_1, a_2, \ldots, a_{n-1}) \rightarrow$ n איברים $\qquad A(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$

$A^{[0]} = (a_0, a_2, a_4, \ldots, a_{n-2})$

$A^{[1]} = (a_1, a_3, a_5, \ldots, a_{n-1})$ $\Big] \rightarrow$ $\frac{n}{2}$ איברים.

$A^{[0]}(x) = a_0 + a_2x + \cdots + a_{n-2}x^{n/2-1}$

$A^{[1]}(x) = a_1 + a_3x + \cdots + a_{n-1}x^{n/2-1}$

$$\boxed{A(x) = A^{[0]}(x^2) + xA^{[1]}(x^2)}$$

$$\left(\boxed{A(\omega_n^0)}, \boxed{A(\omega_n^1)}, \ldots, \boxed{A(\omega_n^{n/2})}, A(\omega_n^{n/2+1}), \ldots, A(\omega_n^{n-1})\right)$$

$$= \left(\boxed{A^{[0]}(\omega_{\frac{n}{2}}^0)}, \boxed{A^{[0]}(\omega_{\frac{n}{2}}^1)}, \ldots, \boxed{A^{[0]}(\omega_{\frac{n}{2}}^{n/2})}\right)$$

$$+ \left(\boxed{A^{[1]}(\omega_{\frac{n}{2}}^0)}, \boxed{A^{[1]}(\omega_{\frac{n}{2}}^1)}, \ldots, \boxed{A^{[1]}(\omega_{\frac{n}{2}}^{n/2})}\right)$$

$$\omega_n^0 \qquad \omega_n^1 \qquad \omega_n^{n/2}$$

תנאי העצירה אם $n=1$

וכך מצאנו $\frac{n}{2}$ איברים.

מהתכונה של הנוסחה והידיעה שהצבנו x^2 אם נציב $-x$ נקבל את הנוסחה הבאה:

$$\boxed{A(-x) = A^{[0]}(x^2) - xA^{[1]}(x^2)}$$

וכך נוכל לשמור על החישוב אך את המכפלה ב-ω_n^i נחליף ב-$(-\omega_n^i)$

כאשר מתקיים מהתכונות של ω (שורשי היחידה) ש:

$-1 = \omega_n^{n/2}$,

$\omega_n^{n/2+1} = -\omega_n^1$

ניתן להסתכל על המעבר כצורה של כפל מטריצות:

$$\overset{F^{-1}}{\begin{pmatrix} \\ \\ \\ \end{pmatrix}} \begin{pmatrix} A(x_0) \\ A(x_1) \\ \vdots \\ A(x_{n-1}) \end{pmatrix} \rightarrow \begin{pmatrix} A(\omega_n^0) \\ A(\omega_n^1) \\ \vdots \\ A(\omega_n^{n-1}) \end{pmatrix} = \overset{F}{\begin{pmatrix} 1 & \omega_n^{0\cdot 1} & \omega_n^{0\cdot 2} & \cdots & \omega_n^{0\cdot(n-1)} \\ 1 & \omega_n^{1\cdot 1} & \omega_n^{1\cdot 2} & & \omega_n^{1\cdot(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{(n-1)\cdot 1} & \omega_n^{(n-1)\cdot 2} & \cdots & \omega_n^{(n-1)(n-1)} \end{pmatrix}} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

כדי לבצע את החישוב ההפוך ולגלות את אגף ימין (את המקדמים)
צריך לכפול בהופכית.

איך נמצא את ההופכית? נחלק ב- n כך: (ונהפוך את החזקה לשלילית)

מטריצה F מוגדרת כך - $d_{ij}=\omega_n^{ij}$

מטריצה F^{-1} מוגדרת כך - $d_{ij}=\frac{\omega_n^{-ij}}{n}$

בעיה: מכפלת 2 פולינומים מדרגה לא שווה.

קלט: שני פולינומים A ו- B. A מדרגה חסומה n ו- B מדרגה חסומה m
כאשר $m < n$.

פלט: תוצאות מכפלת הפולינומים $A\cdot B$.

הצעות אלגוריתמיים:

1) הפתרון הנאיבי - נרפד את B באפסים עד שיהיה בגודל n ונבצע את FFT.
ייקח לנו $O(n\log(n))$

2) נחלק את פולינום A לערך $\frac{n}{m}$ חלקים באורך m (במידת הצורך נרפד את האחרון באפסים), נכפול כל אחד מהחלקים ב- B באמצעות FFT.
נבצע את התוצאות של המכפלה למכפלה כוללת.

פסאודו קוד:

```
FFT_m_n(A,B)
    v ← [0]
    for (i=0 to n/m-1)
        v ← A[i·m ... i·m+m-1]·B  by FFT
        for (j=0 to 2m-2)
            V[im+j] ← V[im+j] + V[j]
    return v;
```

בעיה: התאמת מחרוזות עם don't cares.

התאמת מחרוזות קלסית:

קלט: מחרוזת T באורך n ותבנית P באורך m מעל א"ב Σ.

פלט: המיקומים בטקסט בהם מתחילה ההתאמה של המחרוזת P.

$\Leftarrow$ ניתן לפתור ב-$O(n)$ במקרה הגרוע.

הווריאנט המכונה התאמת מחרוזות עם don't cares, מאפשר את קיומו של תו מיוחד המכונה don't care (או ג'וקר) ומסומן ע"י ϕ שמתאים לכל תו בתבנית או בטקסט.

כלומר, נאמר שיש התאמה של התבנית במקום ה-i בטקסט אם מתקיים:

$$
\begin{array}{ccccc}
T[i]=P[0] & \text{או} & P[0]=\phi & \text{או} & T[i]=\phi \\
T[i+1]=P[1] & \text{או} & P[1]=\phi & \text{או} & T[i+1]=\phi \\
\vdots & & \vdots & & \vdots \\
T[i+m+1]=P[m-1] & \text{או} & P[m-1]=\phi & \text{או} & T[i+m+1]=\phi
\end{array}
$$

נפתור את הבעיה עבור א"ב בינארי $\Sigma=\{0,1\}$

פתרון -

נגדיר תו חדש באופן הבא:
$$a\cdot b=\begin{cases}0 & , a=b\\ 1 & , a\neq b\end{cases}$$

עבור התו המיוחד Φ מתקיים $a\cdot\Phi=\Phi\cdot a=0$ לכל a מהא"ב.

נסתכל על הדוגמא:

$$
\begin{array}{lccccc}
T & \boxed{a_0} & \boxed{a_1} & \boxed{a_2} & \boxed{a_3} & \boxed{a_4} \\
P & & & \boxed{b_0} & \boxed{b_1} & \boxed{b_2}
\end{array}
$$

נבצע חילוק ארוך:

$$
\begin{array}{cc|ccc|cc}
 & & a_0 & a_1 & a_2 & a_3 & a_4 \\
\times & & & & b_2 & b_1 & b_0 \\
\hline
 & & a_0b_0 & a_1b_0 & a_2b_0 & a_3b_0 & a_4b_0 \\
 & a_0b_1 & a_1b_1 & a_2b_1 & a_3b_1 & a_4b_1 & \\
a_0b_2 & a_1b_2 & a_2b_2 & a_3b_2 & a_4b_2 & &
\end{array}
$$

5.12.16

$T = A\,B\,B\,\Phi\,A\,\Phi$

$P = A\,\Phi\,B$

$\} A=0,\ B=1$

×		A	B	B	Φ	A	Φ
					B	Φ	A
		AA	BA	BA	ΦA	AA	ΦA
	AΦ	BΦ	BΦ	$\Phi\Phi$	AΦ	$\Phi\Phi$	
AB	BB	BB	ΦB	AB	ΦB		
		0	1	2	0		

נספור לפי לוח הכפל החדש.

T\P	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

$\overline{T}\cdot P^R$	0	1
1	0	1
0	0	0

$\overline{T}\cdot P^R$:

×		1	0	0	Φ	1	Φ
					1	Φ	0
		0	0	0	0	0	0
	0	0	0	0	0	0	
1	0	0	0	1	0		
		0	0	1	0		

+

$T\cdot\overline{P^R}$:

×		0	1	1	Φ	0	Φ
					0	Φ	1
		0	1	1	0	0	0
	0	0	0	0	0	0	
0	0	0	0	0	0		
		0	1	1	0		