5/5/03

הגדרה - נתונים N מס' $x_0, x_1, \ldots, x_{N-1}$

ואנו מעוניינים למצוא את ה- MAX וה- MIN מה-נתונים.

FFT - חלוקה חכמה של סדרה לתת-סדרות, עם יעילות זמן.

"inplace" - אלגוריתמים שבהם אין צורך להקצות מערכי זיכרון
הנוסף מאחר למערך הנתונים המקורי.

partition - הצבת קו באמצע המערך וחלוקתו לשתי קבוצות.

decimation - עבור כל שני איברים, שולחים אחד ימינה ואחד שמאלה.

- תחום הזמן $x_0, \cancel{x_1}, x_2, \cancel{x_3}, \ldots, x_{N-1}$ partition
- תחום התדר $X_0, X_1, \ldots \mid \cancel{\ldots X_{N-1}}$ decimation

נניח שהיינו מבצעים את ה-partition מחקנו את כל האיברים האי
זוגיים. אילו ח איברים אנו צריכים להוריד בתחום התדר.
ומכיוון שהמקום לשקול כל שנייה בתחום הזמן, הדגימה נעשה
כל שתי שניות. ומכיוון התדר המקסימלי צריך להיות פי 2
ולכן נוריד את כל החצי השני של האיברים בתחום התדר.
לכן דצימציה בזמן שקולה ל-partition בתדר.
כלומר הסקנו לפי משפט הדגימה - היינו צריכים להוריד את כל
החצי איברים אחרונים בסדרה כיוון שבה לא היינו צריכים
אותם לפי משפט הדגימה.

partition $x_0, x_1, x_2, x_3 \mid \cancel{x_4 \; x_5 \; x_6 \; x_7}$

decimation $X_0, \cancel{X_1}, X_2, \cancel{X_3}, X_4, \cancel{X_5}, X_6, \cancel{X_7}$

כאן אנו עושים את ההיפך - partition בתחום הזמן.

עקרון אי הודאות - $\Delta t \Delta \omega \geq 2\pi$

הזמן שמדגמן

התדר שמדגמן

ככל שנתונים לנו זמן קצר יותר למדוד את האות קשה
לנו יותר לגלות את התדר שלו - הסיגנל
הפוך יש לנו פחות בתחום התדר ונותרים האיברים האי-זוגיים.

בעיית הכפלת שני מספרים

$$\begin{array}{cccc} A_3 & A_2 & A_1 & A_0 \\ B_3 & B_2 & B_1 & B_0 \\ \hline A_3B_0 & A_2B_0 & A_1B_0 & A_0B_0 \\ B_1A_3 & B_1A_2 & B_1A_1 & B_1A_0 \\ & \vdots & & \end{array}$$

אנו רואים שהכפלה לוקחת $O(n^2)$ פעולות הכפלה (את החיבור אנו לא סופרים).

- כיצד אנו יכולים להקטין את פעולות ההכפלה ?

נעשה על הצגה partition :

$$A = A_L 2^{N/2} + A_R$$

$$B = B_L 2^{N/2} + B_R$$

$$A \cdot B = (A_L 2^{N/2} + A_R)(B_L 2^{N/2} + B_R)$$

- בדרך הטריוויאלית יש לנו 4 מכפלות:

$$= \underline{A_L B_L} 2^N + (\underline{A_L B_R} + \underline{A_R B_L}) 2^{N/2} + \underline{A_R B_R}$$

- בדרך החדשה יש לנו 3 מכפלות:

$$= \underbrace{A_L B_L}_{N/2 \text{ ביטים}} (2^N + 2^{N/2}) + \underbrace{(A_L - A_R)(B_R - B_L)}_{N/2 \text{ ביטים}} 2^{N/2} + \underbrace{A_R B_R}_{N/2 \text{ ביטים}} (2^{N/2} + 1)$$

- ולכן זמן החישוב : $3\left(\frac{N}{2}\right)^2 = \frac{3}{4} N^2$

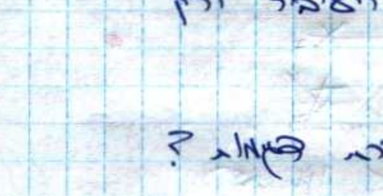
את הטריק הזה ניתן לעשות באופן רקורסיבי ולהרוויח עוד.
אם נעשה $\log_2$ פעמים חלוקה ב- 2 אנו מגיעים לביט אחד ונוכל
לעשות לבד את הפעולה הבסיסית. אלגוריתם כזה קוראים
Toom cook וזו השיטה הכי יעילה להכפלה במחשבים.
היום עובדים כמובן על FFT שהוא הכי מהיר.

- למה אנו רוצים לפתח את אלגוריתם FFT ולא מסתפקים ב- DFT?

Real Time - אנו מקבלים דגימה, מספיקים לעשות את העיבוד ורק
בסיום העיבוד, מקבלים דגימה חדשה.

- מה קורה אם יש לנו אלגוריתם שעובד על סדרת דגימות ?

Double Buffering - משתמשים בבאפר נוסף כך שכאשר אנו
מעבדים את ה- N דגימות שקיבלנו, נוכל לקבל דגימה נוספת
בבאפר הנוסף.



- לכן הזמן שמותר לנו לעבד מדגם אחד הוא עד שהמדגם השני מגיע. לכן יש לנו זמן של N דגימות מהדגם לעשות את העיבוד שאנו רוצים, כלומר - $O(N)$

- מדגם אחד אנו רוצים לעבוד על Buffer (שלם) ולכן צריך להגדיל את התקורה אך מדגם שני כבר יש - Buffer (שלם) ולכן, יקח לנו הרבה יותר זמן לעשות את העיבוד.

- רק אם זמן העיבוד הוא $O(N)$, נרצה להגדיל את ה-Buffer ולהרוויח זמן לעשות את העיבוד - אומנם נצטרך לקחת יותר זמן יותר, אך צריך נרצה לעשות כך.

אלגוריתם - DFT

מס' דגימות של

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk} \qquad W_N = e^{-i\frac{2\pi}{N}} = \cos\frac{2\pi}{N} - i\sin\frac{2\pi}{N}$$

$x_0 \; x_1 \; x_2 \; \dots \; x_{N-1}$

$X_0 \; X_1 \; X_2 \; \dots \; X_{N-1}$

יש לנו $O(N^2)$ מכפלות קומפלקסיות ו $O(N^2)$ חיבורים קומפלקסים.

לכן יש לנו $4O(N^2)$ מכפלות ממשיות ו-$4O(N^2)$ חיבורים.

$$X_0 = x_0 W_4^{0\cdot 0} + x_1 W_4^{1\cdot 0} + x_2 W_4^{2\cdot 0} + x_3 W_4^{3\cdot 0}$$

$$X_1 = x_0 W_4^{0\cdot 1} + x_1 W_4^{1\cdot 1} + x_2 W_4^{2\cdot 1} + x_3 W_4^{3\cdot 1}$$

אנו רואים ש-DSP לא יכול לעבוד עם DFT כיון שהוא חייב real time ו- DFT הוא בזמן ריצה $O(n^2)$ שלא מאפשר לעבוד ב- $O(n^2)$.

- נראה מה אפשר לשפר את ה- DFT:

1. אנו רוצים לחלק את החישוב לשני סדרות:

$$N^2 \Rightarrow \left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 + \text{חישוב איחוד התוצאה} = \frac{N^2}{2} + \text{חישוב איחוד התוצאה}$$

לכן אם נרצה לעשות את חישוב איחוד התוצאה בפחות מ- $\frac{N^2}{2}$ הרווחנו.

נשים לב בדוגמא נוספת של סימטריה.

$$W_N = e^{-\frac{2\pi i}{N}} = \cos\frac{2\pi}{N} - i\sin\frac{2\pi}{N}$$

$$W_N^N = 1$$
$$W_N^{N/2} = -1$$
$$W_N^2 = W_{N/2}$$

חשוב לזכור

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk}$$

אנחנו נמצאים בחלק הימני ב-partition הקודם = $X_{k+N/2}$

$$= \sum_{n=0}^{N-1} x_n W_N^{n(k+N/2)} = \sum_{n=0}^{N-1} (-1)^n x_n W_N^{nk}$$

$$W_N^{n(k+N/2)} \to W_N^{nk+nN/2} \to W_N^{nk}\cdot W_N^{n\cdot N/2} \to W_N^{nk}\cdot(-1)^n$$

- אנו רואים שאנו יכולים לחסוך חצי מהמכפלות שאנו עושים רק על החיבורים שלהם.

האלגוריתם שעשינו עכשיו קוראים Decimation In Time - DIT

אנו עושים כך כיוון שאין סימטריה ב-partition של הסדר.

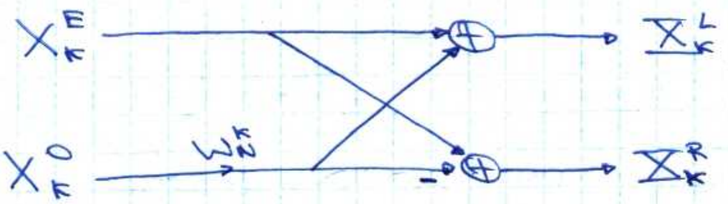
$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk} = \sum_{n=0}^{N/2-1}\left(x_{2n} W_N^{2nk} + x_{2n+1} W_N^{(2n+1)k}\right)$$

$$= \sum_{n=0}^{N/2-1}\left(x_{2n} W_{N/2}^{nk}\right) + \sum_{n=0}^{N/2-1} x_{2n+1} W_{N/2}^{nk} W_N^{k}$$

$$= \underbrace{\sum_{n=0}^{N/2-1} x_n^{E} W_{N/2}^{nk}}_{\text{ה-DFT של הסדרה הזוגית}} + W_N^{k}\underbrace{\sum_{n=0}^{N/2-1} x_n^{O} W_{N/2}^{nk}}_{\text{ה-DFT של הסדרה האי-זוגית}}$$

זוגי   אי-זוגי

עם זאת אנו חוזרים לחשב את $X_{k+N/2}$ אנו מבצעים אותו את אותו החשבון רק עם מינוס



19/5/03

נמשיך משעור קודם:

DFT:
$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk}$$

אמרנו שבמקרה real time אנחנו צריכים לעשות את ההמרה בזמן $O(N)$ אך כאן אנו רואים שההמרה נעשית ב-$O(N^2)$.

תוצאה הכרחית של פונקציה של ה-partition תצריך לנו אם לאחר החלוקה, נרצה לעשות את ההמרה מחדש בפחות מ-$\frac{N^2}{2}$:

$$N^2 \xrightarrow[\text{partition}]{\text{פונקציה של}} \left(\frac{N}{2}\right)^2 + \left(\frac{N}{2}\right)^2 = \cancel{2}\cdot\frac{N^2}{\cancel{4}} = \frac{N^2}{2} \xrightarrow[\text{מחדש}]{\text{ההמרה}}$$

כמובן שפונקציה בתחום הזמן יורדת partition בתחום התדר,

ולכן: $x_0x_2x_4\ldots\ x_1x_3x_5$
$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk}$$

$$X_{k+N/2} = \sum_{n=0}^{N-1} x_n W^{n(k+N/2)} = \sum_{n=0}^{N-1} x_n W_N^{nk}\left(W_N^{N/2}\right)^n = \sum_{n=0}^{N-1}(-1)^n x_n W_N^{nk}$$

נפרק לזוגיים ולאי-זוגיים:

$$X_k = \sum_{n=0}^{N-1} x_n W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} \underbrace{x_{2n} W_N^{2nk}}_{\text{זוגיים}} + \underbrace{x_{2n+1} W_N^{(2n+1)k}}_{\text{אי-זוגיים}}$$

זוגי, אי-זוגי

$$= \sum_{n=0}^{\frac{N}{2}-1} x_n^E W_N^{2nk} + x_n^O W_N^{2nk} W_N^k = \sum_{n=0}^{N/2-1} x_n^E W_{N/2}^{nk} + x_n^O W_{N/2}^{nk} W_N^k$$

לא תלוי ב-n ולכן ניתן להוציא מחוץ ל-Σ

$$= X_k^E + W_N^k X_k^O$$

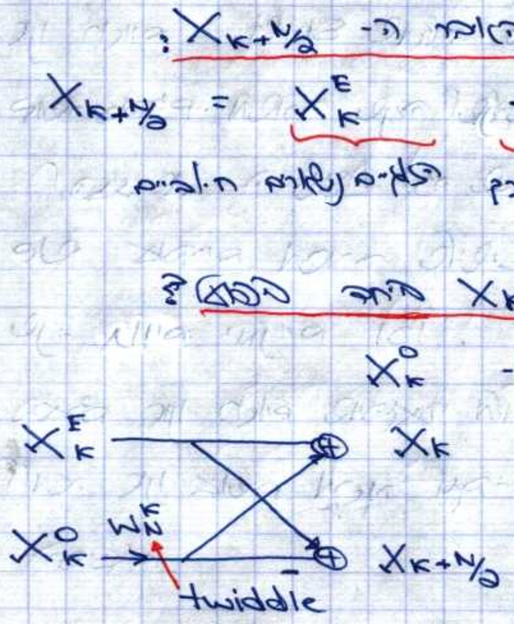
ולפי מה שמצאנו למעלה לגבי הערך ה-$X_{k+N/2}$:

$$X_{k+N/2} = \underbrace{X_k^E}_{\text{הזוגיים נשארים חיוביים}} - \underbrace{W_N^k X_k^O}_{\text{האי-זוגיים מקבלים ערך שלילי}}$$

כיצד מחשבים את X_k ו-$X_{k+N/2}$ יחד בפעם?

נניח שחישבנו כבר את X_k^E ו-X_k^O

פרפר ה-DIT:

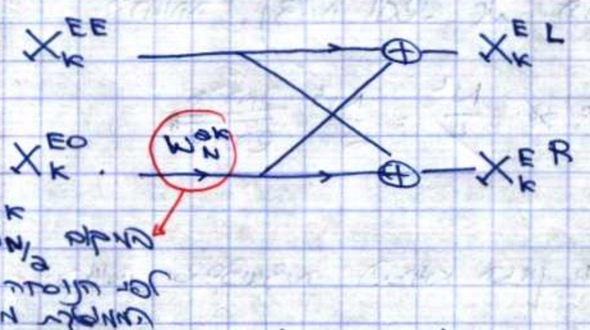


• במקום לעשות n מכפלות n פעמים, אני עושה רק $\frac{n}{2}$ מכפלות
באופן רקורסיבי שכל מס' הפעמים הכולל נראה בהמשך.

• כמובן נמשיך באופן רקורסיבי עד שנגיע לשני איברים שבהם
עושים פעולת חיבור וחיסור (ללא מכפלה).

• נזכור ש: $\boxed{W_{N/2} = W_N^2}$

בשלב זה נניח ש-n הוא חזקה של 2 ונתחיל לבנות רקורסיה:
נרצה לחשב את X_k^E



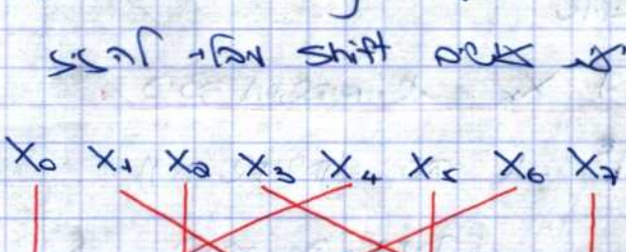
• השרטוט של הרקורסיה שמתבצעת על $x_0, \dots, x_7$
(מתוך הספר הצהוב 543 - 542).

• וכעת אנו יכולים גם לספור את מס' הפעמים הכולל שנצטרך
לעשות את הפעולה הזו - $n \log_2$.
ולכן אנו מקבלים זמן ריצה $\frac{1}{2} \cdot N \log_2 N$ במקום $O(N^2)$.

דוגמא עם 16:

0000	0001	0010	0011	0100				1000	1001	1010					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000	0010	0100						0001	0011	0101					
0	2	4	6	8	10	12	14	1	3	5	7	9	11	13	15
0000	0100							0001	0101	1001					
0	4	8	12	2	6	10	14	1	5	9	13	3	7	11	15
										0101					
0	8	4	12	2	10	6	14	1	9	5	13	3	11	7	15

אנו רואים Shift שמגלה המעבר בין שורה ראשונה לשניה.
כאשר עוברים משורה שניה לשלישית עושים גם כן Shift אך מבלי
להזיז את ה-LSB (זה שקובע האם זוגי - אי זוגי הרי בהתאמה).
כאשר עוברים משורה שלישית לרביעית עושים Shift מבלי להזיז
שני ביטים ימניים וכו'.



בעצם אנו רואים שמתבצע מימוש
חומרה אנו עושים מראה לגבי
הביטים.

- היא להפסת הבדים בדגרת מינקה לין פקטור הכפלה, ומתניגנו כה נשאר אותו זמן ריצה.

- יש הנחות שנמצאים ב- merge sort שווה לאט המתודרים שנמצאים ב- FFT לכפל (שאלה השאלה היום לא ניתן לשפר את ה-$N\log_2 N$. התשובה היא שלא ניתן אם לא יפתח הוכחה לכך.

- בתחום ה- DSP לא היה מתפתח ללא אלגוריתם ה-FFT. נראה עד כמה FFT יעיל בדוגמה!

$N = 1024 = 2^{10}$

$N^2 = (2^{10})^2 = 2^{20}$ - ללא FFT

$\frac{1}{2} N\log_2 N = \frac{1}{2}\cdot 2^{10}\cdot 10 = 5\cdot 2^{10} \approx 2^{12}$ - עם FFT

- כאשר יש לנו אלגוריתם $O(N^2)$ הוא נשאר בגדר real time עבור N-ים קטנים, לעומת $O(N\log N)$ שבו נשארים עבור N-ים גדולים מאוד ולכן הוא נחשב מספיק טוב עבור real time בעולם המעשי למרות שהוא לא $O(N)$.

- מה לגבי חישוב ה-W?

גם אותם אפשר לחשב בעזרת רקורסיה: $W_8^3 = W_8^2 \cdot W_8^1$.

ולפיכך נחשב רק את W_8^1 - אבל הוא שווה ל-1.

- באלגוריתם של גרצל משתמשים בשיטה זו!

- שיטת horner:

$p(x) = \underbrace{a_0 + a_1x + a_2x + a_3x}_{4\ \text{מכפלות}}$

$horner(p(x)) = \underbrace{((a_3x + a_2)x + a_1)x + a_0}_{3\ \text{מכפלות}}$

- כיצד השיטה עוזרת לנו?

$X = \sum_{n=0}^{N-1} x_n (W_N^k)^n$

$= x_0 W_N^0 + x_1 W_N^1 + x_2 W_N^2 + \dots$

- אנו רואים שמה שרשמנו קודם זה היה בצורת פולינום.

נוכל לכתוב את X_k בצורת Horner:

$$X_k = (((x_{N-1}w + x_{N-2})w + x_{N-3})w + \dots + x_0$$

אנו רואים יתרון פה לגבי חישוב ה- W_N^k אך לא לגבי חישוב ה- X-ים.

יש לנו יתרון פה כאשר אנו לא רוצים לחשב את כל האיברים אלא רק איברים ספציפיים. חישוב כל האיברים עדיין ייקח לנו $O(N^2)$ אבל החישוב איבר ספציפי הינו טוב הרבה יותר.

מתי זה יעיל באמת?

	H_1	H_2	H_3	H_4
L_1				
L_2				
L_3				
L_4				

כאשר למשל אנו מחפשים ב- DTMF את האנרגיה של הגל.

אם אנו מחפשים למשל את (L_1, H_3) חבל לחשב את כל ה- FFT. עדיף להשתמש ב- Horner.

- **שתי בעיות להפעלת Horner מעל הנוסחה:**

1. המכפלות הן קומפלקסיות.
2. מתחילים מהאיבר האחרון (צריך לשמור בזיכרון את כל האיברים הקודמים).

- הנוסחה של גרצל מצליחה לפתור את שתי הבעיות יחד:

הרעיון הכללי: נגדיר משתנה חדש שהוא $\frac{1}{w} + w$.

נוכל להפוך את נוסחת ה- Horner בצורה מאוד מתוחכמת כך שיהיו פחות מכפלות קומפלקסיות. בכלל צריך את הסינוסים והקוסינוסים. הסבר לגבי האלגוריתם נמצא בספר בעמ' 563.