אלגוריתמים 1 ־ הרצאה 1: FFT

3 בדצמבר 2025

גיא יער־און

0.1 פעולות של פולינומים

פולינום ניתן לכתיבה כך ־ $P(x) = a_0 + a_1x + a_2x^2 + .. + a_{n-1}x^{n-1} = \sum_{i=0}^{n-1} a_ix^i$.

פולינום זה הוא מדרגה $n-1$, והוא מדרגה חסומה k לכל $k \geq n$. (כלומר $P(x)$ חסום $n, n+1$, $n+2$, $2n+17$ וכו').

1. חיבור/חיסור פולינומים: יהיו $A(x) = \sum_{i=0}^{n-1} a_ix^i, B(x) = \sum_{i=0}^{n-1} b_ix^i$ פולינומים. נגדיר את החיבור/חיסור של $A(x)$ ו$B(x)$ כך ־

$$C(x) = A(x) \pm B(x) = \sum_{i=0}^{n-1}(a_i \pm b_i)x^i$$

נשים לב כי דרגת הפולינום $C(x)$ נשארה זהה לדרגה של $A(x), B(x)$.

2. כפל פולינומים: יהיו $A(x) = \sum_{i=0}^{n-1} a_ix^i, B(x) = \sum_{i=0}^{n-1} b_ix^i$ פולינומים. נגדיר את הכפל של $A(x)$ ו$B(x)$ כך ־

$$C(x) = A(x) \times B(x) = \sum_{i=0}^{2n-2} c_ix^i$$

באשר $c_i = \sum_{j=0}^{i} a_jb_{i-j}$. לפעולה זו קוראים **קונבולוציה.**
נשים לב כי דרגת הפולינום C תהיה $2(n-1) = 2n-2$.

3. חישוב ערך: יהי פולינום $A(x) = \sum_{i=0}^{n-1} a_ix^i$, בהינתן ערך x_0, נרצה לחשב את $A(x_0)$.

0.2 ייצוג של פולינומים

יהי פולינום $A(x) = \sum_{i=0}^{n-1} a_ix^i$. נרצה להראות מספר דרכים לייצג את הפולינום:

0.2.1 ייצוג ע"י מקדמים

נרצה לשמור את המקדמים בלבד של הפולינום. נשתמש במערך ARR בגודל n, ונשמור בתוכו את המקדמים:

$$ARR = (a_0, a_1, ..., a_n)$$

ייצוג זה באמצעות מקדמים נחשב לטוב, קיימת פונקציה חח"ע ועל בין עולם ה"ייצוגים" ל"עולם הפולינומים" - מה הכוונה? לא ייתכן שנקבל ייצוג זהה עבור $A_1(x) \neq A_2(x)$ ולא ייתכן ייצוג שונה עבור $A_1(x) = A_2(x)$.

חיבור/חיסור: יהיו פולינומים $A(x), B(x)$ חסומים מדרגה $n-1$, אזי נרצה לחשב את $C(x) = A(x) \pm B(x)$.

$$\forall_{0 \leq i \leq n-1} : c_i = a_i \pm b_i$$

נשים לב כי בהינתן שיטת המקדמים, לחשב פולינום $C(x)$ הנ"ל יעלה $O(n)$ ע"י חיבור/חיסור זוג הערכים בתאים $A[i], B[i]$ לתוך אחד המערכים. למעשה גם לא נצרך שימוש במקום נוסף. **סה"כ - חיבור/חיסור** $O(n)$.

כפל: לפי הנוסחה שתוארה לעיל:

$$\forall_{0 \leq i \leq n-1} c_i = \sum_{j=0}^{i} a_j b_{i-j}$$

נראה כי זמן לחישוב מקדם c_i בודד יעלה $O(n)$ זמן, ולכן חישוב כלל המקדמים, כלומר **חישוב הכפל יעלה** $O(n^2)$ **זמן**.

חישוב ערך: בהינתן x_0 נרצה לחשב את $A(x_0)$. לא יהיה כאן רעיון מתוחכם - נחשב את $x_0^0, x_0^1, ..., x_0^{n-1}$ ולאחר מכן נחשב את החיבור $a_0 x_0^0 + a_1 x_0^1 + ... a_{n-1} x_0^{n-1}$ וסה"כ **חישוב ערך יעלה** $O(n)$.

0.2.2 ייצוג ע"י נקודות

בהינתן ישר מקביל לאחד הצירים, באמצעות נקודה אחת ניתן לדעת לתאר את הישר. בהינתן שידוע כי הישר הוא קו לינארי ישר $y = mx + b$ אזי בהינתן שתי נקודות ניתן לדעת באופן מדויק את משוואת הישר, בהינתן פרבולה $y = ax^2 + bx + c$ בהינתן שלוש נקודות ניתן לדעת באופן מדויק את משוואת הישר וכן הלאה. באופן כללי, יהא פולינום $A(x) = \sum_{i=0}^{n-1} a_i x^i$ אזי באמצעות n נקודות $(x_0, x_1, ..., x_{n-1})$ ניתן לדעת באופן מדויק ולתאר את הפולינום, **בתנאי שהנקודות שונות אחת מהשנייה.**

סה"כ נייצג את הפולינום A באמצעות הנקודות:

$$(x_0, y_0), (x_1, y_1), ..., (x_{n-1}, y_{n-1})$$

באשר $\forall_{0 \leq i \leq n-1} : y_i = A(x_i)$.

האם הייצוג הזה טוב? האמת שכן, כיצד נראה זאת? צריך להראות שהייצוג הוא חח"ע ועל. כלומר, שקיימת פונקציה חח"ע ועל בין הייצוג לפולינומים.

נשים לב כי $y_i = A(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + ... + a_{n-1} x_i^{n-1}$ לכל $0 \leq i \leq n-1$.

נראה כי ניתן לקבל בסה"כ n משוואות, ולפתור מערכת של n-משוואות, ולמצוא כך את כל ערכי

$a_0, a_1, a_2, ..., a_n$. ולכן סה"כ הייצוג הוא חח"ע ועל. באופן פורמלי יותר - נשים לב כי מערכת המשוואות הנ"ל ניתנת לתיאור בצורה מטריציונית:

$$\begin{pmatrix} y_0 \\ y_1 \\ . \\ . \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} x_0^0 & x_0^1 & x_0^2 & .. & x_0^{n-1} \\ x_1^0 & x_1^1 & x_1^2 & .. & x_1^{n-1} \\ x_2^0 & x_2^1 & x_2^2 & .. & x_2^{n-1} \\ .. & .. & .. & .. & .. \\ x_{n-1}^0 & x_{n-1}^1 & x_{n-1}^2 & .. & x_{n-1}^{n-1} \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ . \\ . \\ a_{n-1} \end{pmatrix}$$

מטריצה זו נקראת **מטריצת ונדרמונדה.** נקרא למטריצה V, לוקטור הימני נקרא $\vec{a}$ ולוקטור השמאלי נקרא $\vec{b}$.

טענה: אם במטריצת ונדרמונדה ערכי $(x_0, ..., x_{n-1})$ כולם שונים זה מזה, אזי מטריצת ונדרמונדה הפיכה.

כיוון שV הפיכה אצלנו, נשים לב כי $\vec{a} = V^{-1}\vec{b}$. סה"כ מצאנו דרך לעבור בין ערכי הנקודות ולקבל את המקדמים של הפולינום, ולכן ייצוג זה הוא חח"ע ועל.

כיצד עוברים בין ייצוג ע"י מקדמים לייצוג ע"י נקודות? ע"י n פעולות של חישוב ערך. נבחר $x_0 \neq x_1 \neq ... \neq x_{n-1}$ ונחשב $A(x_0), ..., A(x_{n-1})$. כמה יעלה מעבר זה? כל חישוב עולה $O(n)$ ולכן סה"כ n חישובים יעלו לנו $O(n^2)$.
כיצד עוברים בין ייצוג ע"י נקודות לייצוג ע"י מקדמים? לפעולה זו יש שם - אינטרפולציה. משתמשים בנוסחת לגראנז':

$$A(x) = \sum_{i=0}^{n-1} y_i \cdot \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)}$$

נשים לב כי חישוב זה יעלה $O(n^2)$ זמן, ולכן גם המעבר השני עולה כמו המעבר הראשון.

1. **חיבור:** יהיו שני פולינומים:
   א. נניח כי שני הפולינומים מיוצגים ע"י אותם נקודות $(x_0, x_1, ..., x_{n-1})$

$$A = (x_0, A(x_0)), (x_1, A(x_1)), ..., (x_{n-1}, A(x_{n-1}))$$

$$B = (x_0, B(x_0)), (x_1, B(x_1)), ..., (x_{n-1}, B(x_{n-1}))$$

נשים לב כי במקרה זה נייצג את פולינום החיבור:

$$C = (x_0, C(x_0)), (x_1, C(x_1)), ..., (x_{n-1}, C(x_{n-1}))$$

באשר $\forall_{0\leq i\leq n-1} C(x_i) = A(x_i) + B(x_i)$. נשים לב כי בדרך זו, חיבור פולינומים עולה $O(n)$ זמן.

ב. שני הפולינומים לא בהכרח מיוצגים ע"י אותם נקודות.

אין דרך קסם". מה שנעשה יהיה לבצע אינטרפולציה, באמצעות נוסחת לגראנז'. נעבור לייצוג ע"י מקדמים של שני הפולינומים, זה יעלה עבור כל פולינום $O(n^2)$. אח"כ נחבר את שני הפולינומים בשיטת המקדמים, מה שיעלה עוד $O(n)$, ואח"כ נמיר חזרה את פולינום החיבור משיטת המקדמים חזרה לשיטת הנקודות, מה שיעלה עוד $O(n^2)$. סה"כ - $O(n^2)$ לחיבור פולינומים.

2. **כפל:** יהיו שני פולינומים.

א. נניח כי שני הפולינומים מיוצגים ע"י אותם נקודות $(x_0, x_1, ..., x_{2n-1})$

$$A = (x_0, A(x_0)), (x_1, A(x_1)), ..., (x_{n-1}, A(x_{2n-1}))$$

$$B = (x_0, B(x_0)), (x_1, B(x_1)), ..., (x_{n-1}, B(x_{2n-1}))$$

נשים לב כי במקרה זה נייצג את פולינום הכפל:

$$C = (x_0, C(x_0)), (x_1, C(x_1)), ..., (x_{n-1}, C(x_{2n-1}))$$

באשר $\forall_{0 \le i \le 2n-1} C(x_i) = A(x_i) \times B(x_i)$. נשים לב כי בדרך זו, **כפל פולינומים עולה $O(n)$ זמן.**

נשים לב כי - הדרגה של פולינום הכפל C היא $2n-2$. כלומר צריך לייצג אותו באמצעות $2n-2$ ערכים. לכן בניגוד למקרים אחרים, כאן דרשנו מA וB להיות מיוצגים ע"י $2n-2$ נקודות. אחרת, לא נוכל לכפול.

ב. שני הפולינומים לא בהכרח מיוצגים ע"י אותם נקודות.

באופן דומה, אין פתרון קסם. נבצע אינטרפולציה. נעבור לשיטת המקדמים, שם נכפול ב$O(n^2)$. אח"כ נשתמש חזרה באינטרפולציה לעבור חזרה לשיטת הנקודות. סה"כ $O(n^2)$ לכפל פולינומים במקרה זה.

3. **חישוב ערך:**

בכל מקרה, צריך לבצע אינטרפולציה ואז לחשב ולכן $O(n^2)$.

הערה: אם אנחנו במקרה בו ערכי הx של שני הפולינומים זהים, והערך שנקראנו לחשב הוא כבר אחד מהערכים שאיתם קיבלנו את הפולינום, כל שיש לעשות בשביל לחשב את הערך הוא לחפש את ערך האיקס הספציפי. מה שיעלה לנו $O(logn)$ בהנחה שהנקודות מסודרות בסדר עולה ביחס לערך האיקס.

0.2.3 סיכום הפעולות בשיטות השונות

סוג השיטה / פעולות	מקדמים	שיטת הנקודות - אותם ערכי x	שיטת הנקודות – לא בהכרח אותם ערכי x
חיבור/חיסור	O(n)	O(n)	O(n^2)
כפל	O(n^2)	O(n): צריך שיהיו 2n-2 ערכים שונים לכל פולינום בשביל שנוכל לכפול.	O(n^2)
חישוב ערך	O(n)	O(n^2)	O(n^2)

0.3 אלגוריתם לכפל פולינומים מהיר: התמרת פורייה FFT

יהיו A, B פולינומים המיוצגים ע"י מקדמים. נרצה לקבל את $C = A \times B$. ראינו שאפשר ב$O(n^2)$. בקורס מבני נתונים", ראינו דרך מעניינת לכפול מטריצות)דומה לפולינומים(באמצעות הפרד ומשול ב$O(n^{log_2 3})$. כעת נרצה למצוא שיטה ב$O(nlogn)$.

מה ראינו עד כה? תקבל מקדמים. תבצע $2n-2$ חישובי ערך, ותעבור לשיטת הנקודות. שם תחשב את הכפל ב$O(n)$ זמן, אח"כ תבצע אינטרפולציה חזרה שתעלה $O(n^2)$ וסיימת. סה"כ עלה לך $O(n^2)$. כעת נראה שיטה, שתאפשר את המעבר הראשון והאחרון ב$O(nlogn)$ זמן, מה שיהפוך את האלגוריתם לזמן $O(nlogn)$. כיצד? נרצה לבחור ערכי x־ים ספציפיים מאוד.

המטרה: נרצה לחשב את $A(x)$ בn ערכים שונים - $x_0, ..., x_n$. מדוע לא $2n-2$? קל להסביר n, ואח"כ קל להכליל את הרעיון וברור שאסימפטוטית זו אותה סיבוכיות.

הנחה: n הוא חזקה של 2. ניתן להתגבר על הנחה זו, באופן שלא יפגע בסיבוכיות, אך בשביל הפשטות המתמטית נניח הנחה זו. **מדוע ניתן להניח זאת? אפשר להוסיף מקדמים של אפס ואז תמיד נקבל חזקה של** 2.

נגדיר את הפולינומים הבאים:

$$A_{even}(x) = a_0 + a_2 x + a_4 x^2 + ... + a_{n-2}^{\frac{n}{2}-1}$$

$$A_{odd}(x) = a_1 + a_3 x + a_5 x^2 + ... + a_{n-1}^{\frac{n}{2}-1}$$

טענה:

$$A_{even}(x^2) + xA_{odd}(x^2) = A(x)$$

$$A_{even}(x^2) - xA_{odd}(x^2) = A(-x)$$

נשים לב כי A_{even}, A_{odd} הם מדרגה חסומה $\frac{n}{2}$, שזה חצי הגדרה החסומה של $A(x)$.

נסיון ראשון: נחשב את A_{odd} בn ערכי x שונים: $x_0^2, ..., x_n^2$. נחשב את A_{even} בn ערכי x שונים: $x_0^2, ..., x_n^2$. ואז נשתמש בנוסחה לעיל כאן בטענה, $A(x) = A_{even}(x^2) + xA_{odd}(x^2)$. נשים לב כי החישוב בסוף עולה $O(n)$ שהרי מחשבים n ערכים, ובכל קריאה אנחנו קוראים ל"2 תתי בעיות" ולכאורה רקורסיבית אפשר לקבל את הנוסחה הבאה $T(n) = 2T(\frac{n}{2}) + n$ ולפי משפט האב, $T(n) = O(nlogn)$.

זה לא עובד. למה?

1. הובטח כי $x_0, ..., x_{n-1}$ שונים. אך מי אמר ש$x_0^2, ..., x_{n-1}^2$ שונים? יתכן כי $x_2 = -10, x_3 = 10$ אך $x_2^2 = x_3^2 = 100$ ואינם שונים.

2. בשימוש בהפרד ומשול, מובטח לך כי סוג תת הבעיה שתקבל יהיה זהה לבעיה המקורית רק על קלט קטן יותר. בבעיה המקורית, נדרשנו לחשב n ערכים של A, באופן שיעלה $O(n)$. נשים לב כי הפולינום חסום מדרגה n בהתחלה, ואז מחשבים בהתאם n ערכים. אח"כ לאחר שמסתכלים על $\frac{n}{2}$, הפולינום חסום מדרגה $\frac{n}{2}$ אבל גם כאן אנו נדרשים לחשב n ערכים שונים. וכן הלאה - **זו לא תת בעיה.**

0.3.1 תכונת הנגדיות החלשה

יהי k חזקה של 2. לסדרת ערכי האיקס: $(x_0, x_1, ..., x_{k-1})$ יש את תכונת הנגדיות החלשה אם אחד מהתנאים הבאים מתקיים:

א. $k = 1$

ב.

$$\forall_{0 \leq j \leq \frac{k}{2}-1} : x_{\frac{k}{2}+j} = -x_j$$

דוגמה. הסדרה $5, -1, 3, 2, -5, 1, -3, -2$ היא בעלת תכונת הנגדיות החלשה, נשים לב שהחצי השני הוא הנגדי של החצי הראשון.

נשים לב - כאשר נעלה את איברי הסדרה בריבוע, נקבל כי החצי הראשון של הסדרה שווה לחצי השני.

נסיון שני: נניח כי מטרת העל החדשה שלנו, היא לחשב את A בn ערכי x שמקיימים את תכונת הנגדיות החלשה. מכאן ש: $\forall_{0 \leq j \leq \frac{n}{2}-1} : (x_{\frac{n}{2}+j})^2 = (-x_j)^2 = (x_j)^2$. לכן, הסדרה הגדולה שלנו היא מורכבת משתי סדרות זהות שבאות אחת אחרי השניה. לכן אם נחשב את A על החצי הראשון, אין צורך לחשב על החצי השני כי קיבלנו זאת בחינם". האלגוריתם החדש:

א. נחשב את A_{odd} ב$\frac{n}{2}$ ערכי x: $(x_0^2, x_1^2, ..., x_{\frac{n}{2}-1}^2)$

ב. נחשב את A_{even} ב$\frac{n}{2}$ ערכי x: $(x_0^2, x_1^2, ..., x_{\frac{n}{2}-1}^2)$

ג. לכל $0 \leq i \leq \frac{n}{2} - 1$ יתקיים: $A(x_i) = A_{even}(x_i^2) + x_iA_{odd}(x_i^2)$

ד. לכל $0 \leq i \leq \frac{n}{2} - 1$)החצי השני של הערכים(נשים לב כי לפי הנגדיות החלשה ומעבר שראינו לעיל מתקיים: $A(x_{\frac{n}{2}+i}) = A(-x_i) = A_{even}(x_i^2) - x_iA_{odd}(x_i^2)$, וסה"כ באמצעות א+ב חישבנו גם את החצי הזה של הערכים בלי לבצע פעולות נוספות.

סה"כ חישבנו את n הערכים הפעם, לכאורה ללא הבעיה שהם יהפכו לשונים, לכאורה באופן רקורסיבי ניתן שוב לטעון $T(n) = 2T(\frac{n}{2}) + n = O(nlogn)$. זה שוב לא עובד -

זה שוב לא תת בעיה! **המטרה שלנו הייתה לחשב סדרת ערכים שמקיימים את תכונת הנגדיות החלשה.** לאחר העלאה בריבוע, הם לא מקיימים את תכונת הנגדיות החלשה. ואי אפשר לומר שזו תת בעיה.

0.3.2 תכונת הנגדיות החזקה

יהי k חזקה של 2. לסדרת ערכי האיקס: $(x_0, x_1, ..., x_{k-1})$ יש את תכונת הנגדיות החזקה אם אחד מהתנאים הבאים מתקיים:

א. $k = 1$

ב. לסדרה יש את תכונת הנגדיות החלשה, וגם לסדרה $(x_0^2, x_1^2, ..., x_{\frac{n}{2}-1}^2)$ יש את תכונת הנגדיות החזקה (באופן רקורסיבי).

כעת נשים לב ־ כי הבעיה שהייתה לנו מקודם נפתרה לגמרי. להלן האלגוריתם:

המטרה: לחשב את A מדרגה חסומה n בn ערכי x שונים שמקיימים את תכונת הנגדיות החזקה.

א. נחשב את A_{odd} שהוא מדרגה חסומה $\frac{n}{2}$, ב$\frac{n}{2}$ ערכי x: $(x_0^2, x_1^2, ..., x_{\frac{n}{2}-1}^2)$ ־ מההגדרה הרקורסיבית, ערכים אלו מקיימים את תכונת הנגדיות החזקה.

ב. נחשב את A_{even} שהוא מדרגה חסומה $\frac{n}{2}$, ב$\frac{n}{2}$ ערכי x: $(x_0^2, x_1^2, ..., x_{\frac{n}{2}-1}^2)$ ־ מההגדרה הרקורסיבית, ערכים אלו מקיימים את תכונת הנגדיות החזקה.

ג. לכל $0 \leq i \leq \frac{n}{2} - 1$ יתקיים: $A(x_i) = A_{even}(x_i^2) + x_i A_{odd}(x_i^2)$

ד. לכל $0 \leq i \leq \frac{n}{2} - 1$)החצי השני של הערכים(נשים לב כי לפי הנגדיות החלשה ומעבר שראינו לעיל מתקיים: $A(x_{\frac{n}{2}+i}) = A(-x_i) = A_{even}(x_i^2) - xA_{odd}(x_i^2)$, וסה"כ באמצעות א+ב חישבנו גם את החצי הזה של הערכים בלי לבצע פעולות נוספות.

סה"כ הבעיות נפתרו ־ בכל שלב אנחנו מקבלים תת בעיה, וכן הערכים תמיד יקיימו את תכונת הנגדיות החזקה. סה"כ סיבוכיות הזמן של האלגוריתם הינה $T(n) = 2T(\frac{n}{2}) + O(n)$, וממאסטר נקבל $O(nlogn)$ למעבר ממקדמים לייצוג ע"י נקודות.

0.3.3 איזה מספרים מקיימים את תכונת הנגדיות החזקה?

מספרים מרוכבים. נזכר כי מס' מרוכב ניתן לייצג ע"י $z = a + bi = rcis\Theta = re^{i\Theta}$. אנחנו נתמקד במספרים בהם $r = 1$.

מספר ω נקרא שורש היחידה המרוכב מסדר n, אם $\omega^n = 1$. למשל, נראה כי $z = e^{\frac{2\pi i}{8}}$, מתקיים כי $z^8 = 1$.

נגדיר את המספר ω_n להיות: $\omega_n = e^{\frac{2\pi i}{n}}$. נשים לב כי תמיד $\omega_n^n = 1$. (הערה ־ נשים לב כי $\omega_4 = e^{\frac{\pi i}{2}} = i$. מדוע? נראה כי הזווית היא $\frac{\pi}{2}$, כלומר 90 מעלות. אם נזוז 90 מעלות מהכיוון החיובי של הציר הממשי, נגיע בדיוק למספר i. וכך בדיוק מחשבים מספרים אלו).

n שורשי היחידה מסדר n הינם: $\omega_n^0, \omega_n^1, ..., \omega_n^{n-1}$.

טענה: לכל $0 \leq k \leq n - 1$ מתקיים כי $(w_n)^k$ הוא שורש יחידה מסדר n.

הוכחה: נרצה להוכיח כי מספר זה בחזקת n שווה לאחד. ובכן ־

$$((w_n)^k)^n = ((e^{\frac{2\pi i}{n}})^k)^n = (e^{\frac{2\pi ink}{n}}) = e^{2\pi ik} = e^0 = 1$$

טענה 2: יהי $n > 1$ חזקה של 2. אזי לכל $0 \leq k \leq \frac{n}{2} - 1$ מתקיים $w_n^{\frac{n}{2}+k} = -w_n^k$.

הוכחה:

$$\omega^{\frac{n}{2}+k} = \omega^{\frac{n}{2}} \times \omega^k = \omega^k \times e^{\frac{2\pi\frac{n}{2}}{n}} = -\omega^k$$

טענה 3: יהי $n > 1$, חזקה של 2. אזי הריבועים של $\omega_n^0, \omega_n^1, ..., \omega_n^{n-1}$ הם בדיוק $\frac{n}{2}$ שורשי היחידה מסדר $\frac{n}{2}$. (כלומר, נעלה אותם בריבוע, נקבל בדיוק את שורשי היחידה מסדר $\frac{n}{2}$, וכל אחד מהם יופיע פעמיים - כלומר יהיו כפילויות).

הוכחה: עבור $0 \leq k \leq \frac{n}{2} - 1$ מתקיים

$$(\omega_n^k)^2 = e^{i\frac{2\pi}{n}2k} = (e^{\frac{i2\pi}{\frac{n}{2}}})^k = \omega_{\frac{n}{2}}^k$$

טענה 4: יהי $n \geq 1$ חזקה של 2. סדרת n שורשי היחידה מסדר n: $\omega_n^0, \omega_n^1, \omega_n^2, ..., \omega_n^{n-1}$ מקיימים את תכונת הנגדיות החזקה.

נשים לב - n שורשי היחידה מסדר n הם אכן שונים זה מזה.

0.4 האלגוריתם FFT

המטרה: לקחת את המערך עם המקדמים של A, ולחשב את הפולינום A בn שורשי היחידה מסדר n (אשר הוכחנו שמקיימים את תכונת הנגדיות החזקה). כלומר לחשב את $A(\omega_n^0), ..., A(\omega_n^{n-1})$.

קלט: $A = (a_0, a_1, ..., a_{n-1})$ (מערך המקדמים של הפולינום A) מדרגה חסומה n

א. אם $n = 1$, החזר a_0.

ב. כעת נגדיר את $A_{even} = (a_0, a_2, ..., a_{n-2})$ מדרגה חסומה $\frac{n}{2}$

ג. כעת נגדיר את $A_{odd} = (a_1, a_3, ..., a_{n-1})$ מדרגה חסומה $\frac{n}{2}$

ד. כעת נתחיל את הרקורסיה: $P_{even} = FFT(A_{even})$, כאשר P_{even} יפעיל את FFT על A_{even} - כלומר מחשבים את A_{even} בn שורשי היחידה מסדר $\frac{n}{2}$.
כלומר, מה שחוזר מהרקורסיה הינו $P_{even} = [A_{even}(\omega_{\frac{n}{2}}^0), A_{even}(\omega_{\frac{n}{2}}^1), ..., A_{even}(\omega_{\frac{n}{2}}^{\frac{n}{2}-1})]$

ה. בדומה - $P_{odd} = FFT(A_{odd})$

ו. החל מ$j = 0$ עד $j = \frac{n}{2} - 1$ בצע: (כעת אנחנו רוצים לחשב את הפלט שלנו - מחזירים לבסוף וקטור $\vec{y}$ עם חישוב הערכים בהתאם לפי הנוסחאות שראינו)

1. $y_j = P_{even}[j] + w_n^j \times P_{odd}[j]$
(נשים לב כי $y_j = A(\omega_n^j)$ ולפי נוסחה שראינו $A(x) = A_{even}(x^2) + xA_{odd}(x^2)$, כמו כן ניתן להמירה בהתאם $A(\omega_n^j) = A_{even}({\omega_n^j}^2) + \omega_n^j A({\omega_n^j}^2)$, וכפי שראינו מתקיים $(\omega_n^j)^2 = w_{\frac{n}{2}}^j$ לפי טענה3 לעיל, ולכן $A(\omega_n^j) = A_{even}(w_{\frac{n}{2}}^j) + \omega_n^j A(w_{\frac{n}{2}}^j)$, כלומר אם נסתכל על מערכי הקלט ששמרנו זה ממש שקול ל$(P_{even}[j] + w_n^j \times P_{odd}[j]$.

2. $y_{\frac{n}{2}+j} = P_{even}[j] - w_n^j \times P_{odd}[j]$

ז. כשהרקורסיה נגמרת ־ החזר את $(y_0, ..., y_{n-1})$

סיבוכיות זמן הריצה: הנוסחה ־ $T(n) = 2T(\frac{n}{2}) + O(n) = O(nlogn)$, ולכן סה"כ זמן חישוב האלגוריתם שמקבל פולינום בשיטת המקדמים, וממיר אותם לn נקודות ספציפיות (שורשי היחידה מסדר n) לפי שיטת הנקודות הוא $O(nlogn)$.

0.5 כיצד נעבור כעת משיטת הנקודות חזרה לשיטת המקדמים?

נשים לב כי אנחנו יודעים את ערכי x הנקודות שלנו, הם n שורשי היחידה מסדר n.

נחזור ל**מטריצת ונדרמונדה.** נציב $x_0 = (\omega_n)^0 = 1, x_1 = (\omega_n^1), ..., x_{n-1} = (\omega_n^{n-1})$

$$\begin{pmatrix} y_0 \\ y_1 \\ . \\ . \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & .. & 1 \\ 1 & \omega_n^1 & (\omega_n^1)^2 & .. & (\omega_n^1)^{n-1} \\ 1 & (\omega_n^2)^1 & (\omega_n^2)^2 & .. & (\omega_n^2)^{n-1} \\ .. & .. & .. & .. & .. \\ 1 & (\omega_n^{n-1})^1 & (\omega_n^{n-1})^2 & .. & (\omega_n^{n-1})^{n-1} \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ . \\ . \\ a_{n-1} \end{pmatrix}$$

נשים לב כי בידינו קודם לכן היה $\vec{a}$, כפלנו אותו במטריצת הFFT שלנו, V וקיבלנו את $\vec{y}$. באופן כללי ־ כפל נאיבי של מטריצה בוקטור עולה $O(n^2)$ זמן.

מסקנה חשובה (!!)**:** כפל של מטריצת ונדרמונדה שמוגדרת ע"י n מספרים שמקיימים את תכונת הנגדיות החזקה, בוקטור $\vec{a}$ עולה $O(nlogn)$ (שזה בדיוק אותו תהליך שעשה האלגוריתם).

טענה: המטריצה ההופכית של V, FFT^{-1} הינה המטריצה:

$$V^{-1} = FFT^{-1} = \frac{1}{n} \begin{pmatrix} 1 & 1 & 1 & .. & 1 \\ 1 & \omega_n^{-1} & (\omega_n^{-1})^2 & .. & (\omega_n^{-1})^{n-1} \\ 1 & (\omega_n^{-2})^1 & (\omega_n^{-2})^2 & .. & (\omega_n^{-2})^{n-1} \\ .. & .. & .. & .. & .. \\ 1 & (\omega_n^{-(n-1)})^1 & (\omega_n^{-(n-1)})^2 & .. & (\omega_n^{-(n-1)})^{n-1} \end{pmatrix}$$

אם נסתכל על המטריצה $n \times FFT^{-1}$ (שהרי נרצה להכפיל בn כי נשים לב ל$\frac{1}{n}$ שיצא החוצה מהמטריצה), נראה כי היא מטריצת ונדרמונדה על הערכים:
$(\omega_n^0, \omega_n^{-1}, \omega_n^{-2}, ..., \omega_n^{n-1})$.

מסקנה: על מנת לעבור מוקטור $\vec{y}$ לוקטור המקדמים $\vec{a}$, נראה כי בהכפלה במטריצה ההופכית מקבלים ממש $\vec{a} = FFT^{-1} \times \vec{y}$, זו מכפלה של מטריצת ונדרמונדה על n מספרים שמקיימים את תכונת הנגדיות החזקה (שגם הם, n שורשי היחידה מסדר n) , בוקטור, ראינו בטענה לעיל שמכפלה זו עולה $O(nlogn)$, ולכן **המסקנה שלנו היא שגם המעבר חזרה ־ משיטת הנקודות חזרה אל שיטת המקדמים, עולה גם הוא $O(nlogn)$.**

סיכום ־ כפל פולינומים:

א. מקבלים את הפולינומים $A(x), B(x)$ המיוצגים ע"י מקדמים.

ב. בעזרת אלגוריתם FFT, בזמן $O(nlogn)$ מקבלים את $A(x), B(x)$ מיוצגים ע"י n נקודות (שהם שורשי היחידה מסדר n)

ג. מכפילים את שני הפולינומים בזמן $O(n)$ בשיטת הנקודות, כיוון שהם מיוצגים ע"י אותם ערכי x (שורשי היחידה).

ד. משתמשים שוב בFFT, באמצעות הכפלה ע"י המטריצה FFT^{-1} שגם היא מטריצת ונדרמונדה, מחזירים את הפולינומים לשיטת המקדמים, מה שיעלה עוד $O(nlogn)$.

סה"כ - $O(nlogn)$ לכפל פולינומים.